Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes

Ohad Shamir   OHADSH@MICROSOFT.COM

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

Tong Zhang   TZHANG@STAT.RUTGERS.EDU

Department of Statistics, Rutgers University, Piscataway NJ 08854, USA

Abstract

Stochastic Gradient Descent (SGD) is one of the simplest and most popular stochastic optimization methods. While it has already been theoretically studied for decades, the classical analysis usually required non-trivial smoothness assumptions, which do not apply to many modern applications of SGD with non-smooth objective functions such as support vector machines. In this paper, we investigate the performance of SGD *without* such smoothness assumptions, as well as a running average scheme to convert the SGD iterates to a solution with optimal optimization accuracy. In this framework, we prove that after T rounds, the suboptimality of the *last* SGD iterate scales as $\mathcal{O}(\log(T)/\sqrt{T})$ for non-smooth convex objective functions, and $\mathcal{O}(\log(T)/T)$ in the non-smooth strongly convex case. To the best of our knowledge, these are the first bounds of this kind, and almost match the minimax-optimal rates obtainable by appropriate averaging schemes. We also propose a new and simple averaging scheme, which not only attains optimal rates, but can also be easily computed on-the-fly (in contrast, the suffix averaging scheme proposed in Rakhlin et al. (2011) is not as simple to implement). Finally, we provide some experimental illustrations.

1. Introduction

This paper considers one of the simplest and most popular stochastic optimization algorithms, namely

Stochastic Gradient Descent (SGD). SGD can be used to optimize any convex function F over a convex domain $\mathcal{W}$, given access only to unbiased estimates of F's gradients (or more generally, subgradients[1]). This feature makes it very useful for learning problems, where our goal is to minimize generalization error based only on a finite sampled training set. Moreover, SGD is extremely simple and highly scalable, making it particularly suitable for large-scale learning problems.

The algorithm itself proceeds in rounds, and can be described in just a few lines: We initialize $\mathbf{w}_1 \in \mathcal{W}$ (following common practice, we will assume $\mathbf{w}_1 = \mathbf{0}$). At round $t = 1, 2, \ldots$, we obtain a random estimate $\hat{\mathbf{g}}_t$ of a subgradient $\mathbf{g}_t \in \partial F(\mathbf{w}_t)$ so that $\mathbb{E}\hat{\mathbf{g}}_t = \mathbf{g}_t$, and update the iterate $\mathbf{w}_t$ as follows:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t \hat{\mathbf{g}}_t),$$

where η_t is a suitably chosen step-size parameter, and $\Pi_{\mathcal{W}}$ denotes projection on $\mathcal{W}$.

In terms of theoretical analysis, SGD has been studied for decades (for instance, see Kushner & Yin (2003) and references therein), but perhaps surprisingly, there are still important gaps left in our understanding of this method. First of all, most classical results look at *asymptotic* convergence rates, which do not apply to a fixed iteration budget T. In recent years, more attention has been devoted to non-asymptotic bounds (e.g., Bach & Moulines (2011)). However, these classical convergence bounds often make non-trivial smoothness assumptions on the function F, such as Lipschitz-continuity of the gradient or higher-order derivatives. In modern applications, these assumptions often do not hold. For example, if SGD is used to solve the support-vector machine optimization problem (with the standard non-smooth hinge-loss) on a finite training set, then the underlying objective function F is

---

[1]Following a common convention, we still refer to the algorithm in this case as "gradient descent".

essentially *non-smooth*, even at the optimal solution. In general, for machine learning applications F may be non-smooth whenever one uses a non-smooth loss function, and thus a smoothness-based analysis is not appropriate.

Without assuming smoothness, most of the existing analysis has been carried out in the context of online learning - a more difficult setting than our stochastic setting, where the subgradients are assumed to be provided by an adversary. Using online-to-batch conversion, it is possible to show that after T iterations, the average of the iterates, $(\mathbf{w}_1 + \ldots + \mathbf{w}_T)/T$, has $\mathcal{O}(\log(T)/T)$ optimization error for strongly-convex F (see precise definition in Sec. 2), and $\mathcal{O}(1/\sqrt{T})$ error for general convex F (Zinkevich, 2003; Hazan et al., 2007; Hazan & Kale, 2011). However, (Rakhlin et al., 2011) showed that simple averaging is provably suboptimal in a stochastic setting. Instead, they proposed averaging the last αT iterates of SGD (where $\alpha \in (0,1)$, e.g. 1/2), and showed that this averaging scheme has an optimal $\mathcal{O}(1/T)$ convergence rate. In comparison, in the non-smooth setting, there are $\Omega(1/\sqrt{T})$ and $\Omega(1/T)$ lower bounds for convex and strongly-convex problems, respectively (Agarwal et al., 2012).

These results leave open several issues. First, they pertain to averaging significant parts of the iterates, although in practice averaging just over the last few iterates, or returning the last iterate $\mathbf{w}_T$, often works quite well (e.g. (Shalev-Shwartz et al., 2011)). Unless F is smooth, the previous results cannot say much about the optimization error of individual iterates. For example, the results in (Rakhlin et al., 2011) only imply an $\mathcal{O}(1/\sqrt{T})$ convergence rate for the last iterate $\mathbf{w}_T$ with strongly-convex functions, and we are not aware of any results for the last iterate $\mathbf{w}_T$ in the general convex case. In fact to the best of our knowledge, even for the simpler (non-stochastic) gradient descent method (where $\hat{\mathbf{g}}_t = \mathbf{g}_t$), we do not know any existing results that can guarantee the performance of each individual iterate $\mathbf{w}_T$. Second, the theoretically optimal suffix-averaging scheme proposed in (Rakhlin et al., 2011) has some practical limitations, since it cannot be computed on-the-fly: unless we can store all iterates $\mathbf{w}_1, \ldots, \mathbf{w}_T$ in memory, one needs to know the stopping time T beforehand, in order to know when to start computing the suffix average. In practice, T is often not known in advance. This can be partially remedied with a so-called *doubling trick*, but it is still not a simple or natural procedure compared to just averaging all iterates, and the latter was shown to be suboptimal in (Rakhlin et al., 2011).

In this paper, we investigate the convergence rate of SGD and the averaging schemes required to obtain them, with the following contributions:

- We prove that the expected optimization error of every individual iterate $\mathbf{w}_T$ is $\mathcal{O}(\log(T)/T)$ for strongly-convex F, and $\mathcal{O}(\log(T)/\sqrt{T})$ for general convex F without smoothness assumptions on F. These results show that the suboptimality of the last iterate is not much worse than the optimal rates obtainable by averaging schemes, and partially addresses an open problem posed in (Shamir, 2012). Moreover, the latter result is (to the best of our knowledge) the first finite-sample bound on individual iterates of SGD for non-smooth convex optimization. The proof relies on a technique to reduce results on averages of iterates to results on individual iterates, which was implicitly used in (Zhang, 2004) for a somewhat different setting.

- We improve the existing expected error bound on the suffix averaging scheme of (Rakhlin et al., 2011), from $\mathcal{O}((1 + \log(\frac{1}{1-\alpha}))/\alpha T)$ to $\mathcal{O}(\log(\frac{1}{\min\{\alpha,1-\alpha\}})/T)$.

- We propose a new and very simple running average scheme, called *polynomial-decay averaging*, and prove that it enjoys optimal rates of convergence. Unlike suffix-averaging, this new running average scheme can be easily computed on-the-fly.

- We provide a simple experimental study of the averaging schemes discussed in the paper.

We emphasize that although there exist other algorithms with $\mathcal{O}(1/T)$ convergence rate in the strongly convex case (e.g. (Hazan & Kale, 2011; Ouyang & Gray, 2012)), our focus in this paper is on the simple and widely-used SGD algorithm.

2. Preliminaries

We use bold-face letters to denote vectors. We let F denote a convex function over a (closed) convex domain $\mathcal{W}$, which is a subset of some Hilbert space with an induced norm $\| \cdot \|$. We assume that F is minimized at some $\mathbf{w}^* \in \mathcal{W}$. Besides general convex F, we will also consider the important sub-class of strongly-convex functions. Formally, we say that a function F is λ-*strongly convex*, if for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ and any subgradient $\mathbf{g}$ of F at $\mathbf{w}$, it holds that

$$F(\mathbf{w}') \geq F(\mathbf{w}) + \langle \mathbf{g}, \mathbf{w}' - \mathbf{w} \rangle + \frac{\lambda}{2}\|\mathbf{w}' - \mathbf{w}\|^2,$$

where $\lambda > 0$. For a general convex function, the above inequality can always be satisfied with $\lambda = 0$.

As discussed in the introduction, we consider the first-order stochastic optimization setting, where instead of having direct access to F, we only have access to an oracle, which given some $\mathbf{w} \in \mathcal{W}$, returns a random vector $\hat{\mathbf{g}}$ such that $\mathbb{E}[\hat{\mathbf{g}}] \in \partial F(\mathbf{w})$. Our goal is to use a bounded number T of oracle calls, and compute some $\bar{\mathbf{w}} \in \mathcal{W}$ such that the optimization error, $F(\bar{\mathbf{w}}) - F(\mathbf{w}^*)$, is as small as possible. It is well-known that this framework can be applied to learning problems (see for instance (Shalev-Shwartz et al., 2009)): given a hypothesis class $\mathcal{W}$ and a set of T i.i.d. examples, we wish to find a predictor $\mathbf{w}$ whose expected loss $F(\mathbf{w})$ is close to optimal over $\mathcal{W}$. Since the examples are chosen i.i.d., the subgradient of the loss function with respect to any individual example can be shown to be an unbiased estimate of a subgradient of F. We will mostly consider bounds on the expected error (over the oracle's and algorithm's randomness) for simplicity, although it is possible to obtain high-probability bounds in some cases.

In terms of the step-size η_t in the strongly-convex case, we will generally assume it equals $1/(\lambda t)$. We note that this is without much loss of generality, since if the step size is $c/\lambda t$ for some $c \geq 1$, then it is equivalent to taking step sizes $1/(\lambda' t)$ where $\lambda' := \lambda/c \leq \lambda$ is a lower-bound on the strong convexity parameter. Since any λ-strongly convex function is also λ'-strongly convex, we can use the analysis here to get upper bounds in terms of λ', and if so desired, substitute λ/c instead of λ' in the final bound.

When we run SGD, we let $\hat{\mathbf{g}}_t$ denote the random vector obtained at round t (when we query at $\mathbf{w}_t$), and let $\mathbf{g}_t = \mathbb{E}[\hat{\mathbf{g}}_t]$ denote the underlying subgradient of F. To facilitate our convergence bounds, we assume that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|^2] \leq G^2$ for some fixed G. Also, when optimizing general convex functions, we will assume that the diameter of $\mathcal{W}$, namely $\sup_{\mathbf{w},\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|$, is bounded by some constant D.

3. Convergence of Individual SGD Iterates

We begin by considering the case of strongly convex F, and prove the following bound on the expected error of any individual iterate $\mathbf{w}_T$. In this theorem as well as later ones, we did not attempt to optimize constants.

Theorem 1. *Suppose F is λ-strongly convex, and that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|^2] \leq G^2$ for all t. Consider SGD with step sizes*

$\eta_t = 1/\lambda t$. *Then for any $T > 1$, it holds that*

$$\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \leq \frac{17G^2(1 + \log(T))}{\lambda T}.$$

Proof. The beginning of the proof is standard. By convexity of $\mathcal{W}$, we have the following for any $\mathbf{w} \in \mathcal{W}$:

$$
\begin{aligned}
\mathbb{E}\left[\|\mathbf{w}_{t+1} - \mathbf{w}\|^2\right] &= \mathbb{E}[\|\Pi_\mathcal{W}(\mathbf{w}_t - \eta_t \hat{\mathbf{g}}_t) - \mathbf{w}\|^2] \\
&\leq \mathbb{E}\left[\|\mathbf{w}_t - \eta_t \hat{\mathbf{g}}_t - \mathbf{w}\|^2\right] \\
&\leq \mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}\|^2\right] - 2\eta_t \mathbb{E}[\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}\rangle] + \eta_t^2 G^2.
\end{aligned}
$$

Let k be an arbitrary element in $\{1, \ldots, \lfloor T/2 \rfloor\}$. Extracting the inner product, summing over all $t = T - k, \ldots, T$, and rearranging, we get

$$
\begin{aligned}
\sum_{t=T-k}^{T} \mathbb{E}[\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}\rangle] &\leq \frac{1}{2\eta_{T-k}} \mathbb{E}[\|\mathbf{w}_{T-k} - \mathbf{w}\|^2] \\
&+ \sum_{t=T-k+1}^{T} \frac{\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2]}{2}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \frac{G^2}{2}\sum_{t=T-k}^{T} \eta_t.
\end{aligned}
\tag{1}
$$

By convexity of F, we can lower bound $\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}\rangle$ by $F(\mathbf{w}_t) - F(\mathbf{w})$. Plugging this in and substituting $\eta_t = 1/\lambda t$, we get

$$
\begin{aligned}
\mathbb{E}\left[\sum_{t=T-k}^{T}(F(\mathbf{w}_t) - F(\mathbf{w}))\right] &\leq \frac{\lambda(T-k)}{2}\mathbb{E}[\|\mathbf{w}_{T-k} - \mathbf{w}\|^2] \\
&+ \frac{\lambda}{2}\sum_{t=T-k+1}^{T}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2] + \frac{G^2}{2\lambda}\sum_{t=T-k}^{T}\frac{1}{t}.
\end{aligned}
\tag{2}
$$

Now comes the crucial trick: instead of picking $\mathbf{w} = \mathbf{w}^*$, as done in standard analysis ((Hazan et al., 2007; Rakhlin et al., 2011)), we instead pick $\mathbf{w} = \mathbf{w}_{T-k}$. We also use the fact that $\mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}^*\|^2\right] \leq \frac{4G^2}{\lambda^2 t}$ ((Rakhlin et al., 2011), Lemma 1), which implies that for any $t \geq T - k$,

$$
\begin{aligned}
&\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_{T-k}\|^2] \\
&\leq 2\mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \|\mathbf{w}_{T-k} - \mathbf{w}^*\|^2\right] \\
&\leq \frac{8G^2}{\lambda^2}\left(\frac{1}{t} + \frac{1}{T-k}\right) \leq \frac{16G^2}{\lambda^2(T-k)} \leq \frac{32G^2}{\lambda^2 T}.
\end{aligned}
$$

Plugging this back into Eq. (2), we get

$$\mathbb{E}\left[\sum_{t=T-k}^{T}(F(\mathbf{w}_t) - F(\mathbf{w}_{T-k}))\right] \leq \frac{16G^2 k}{\lambda T} + \frac{G^2}{2\lambda}\sum_{t=T-k}^{T}\frac{1}{t}.$$

Let $S_k = \frac{1}{k+1}\sum_{t=T-k}^{T}\mathbb{E}[F(\mathbf{w}_t)]$ be the expected average value of the last $k + 1$ iterates. The bound above implies that

$$-\mathbb{E}[F(\mathbf{w}_{T-k})] \leq -\mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{T} + \sum_{t=T-k}^{T}\frac{1}{(k+1)t}\right).$$

By the definition of S_k and the inequality above, we have

$$k\mathbb{E}[S_{k-1}] = (k+1)\mathbb{E}[S_k] - \mathbb{E}[F(\mathbf{w}_{T-k})]$$

$$\leq (k+1)\mathbb{E}[S_k] - \mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{T} + \sum_{t=T-k}^{T}\frac{1}{(k+1)t}\right),$$

and dividing by k, implies

$$\mathbb{E}[S_{k-1}] \leq \mathbb{E}[S_k] + \frac{G^2}{2\lambda}\left(\frac{32}{kT} + \sum_{t=T-k}^{T}\frac{1}{k(k+1)t}\right). \quad (3)$$

Using this inequality repeatedly and summing from $k=1$ to $k = \lfloor T/2 \rfloor$, we have

$$\mathbb{E}[F(\mathbf{w}_T)] = \mathbb{E}[S_0] \leq \mathbb{E}[S_{\lfloor T/2 \rfloor}] + \frac{16G^2}{\lambda T}\sum_{k=1}^{\lfloor T/2 \rfloor}\frac{1}{k}$$

$$+ \frac{G^2}{2\lambda}\sum_{k=1}^{\lfloor T/2 \rfloor}\sum_{t=T-k}^{T}\frac{1}{k(k+1)t}. \quad (4)$$

It now just remains to bound these terms. $\mathbb{E}[S_{T/2}]$ is the expected average value of the last $\lfloor T/2 \rfloor$ iterates, which was already analyzed in ((Rakhlin et al., 2011), Theorem 5), yielding a bound of

$$\mathbb{E}[S_{\lfloor T/2 \rfloor}] \leq F(\mathbf{w}^*) + \frac{10G^2}{\lambda T}$$

for $T > 1$. Moreover, we have $\sum_{k=1}^{\lfloor T/2 \rfloor}(1/k) \leq 1 + \log(T/2)$. Finally, we have

$$\sum_{k=1}^{\lfloor T/2 \rfloor}\sum_{t=T-k}^{T}\frac{1}{k(k+1)t} \leq \sum_{k=1}^{\lfloor T/2 \rfloor}\frac{1}{k(T-k)}$$

$$= \frac{1}{T}\sum_{k=1}^{\lfloor T/2 \rfloor}(\frac{1}{k} + \frac{1}{T-k}) \leq (1 + \log(T))/T.$$

The result follows by substituting the above bounds into Eq. (4) and simplifying for readability. $\quad\square$

Using a similar technique, we can also get an individual iterate bound, in the case of a general convex function F that may be non-smooth. We note that a similar technique was used in (Zhang, 2004), but for a different algorithm (one with constant learning rate), and the result was less explicit.

Theorem 2. *Suppose that F is convex, and that for some constants D, G, it holds that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|] \leq G^2$ for all t, and $\sup_{\mathbf{w},\mathbf{w}' \in \mathcal{W}}\|\mathbf{w} - \mathbf{w}'\| \leq D$. Consider SGD with step sizes $\eta_t = c/\sqrt{t}$ where $c > 0$ is a constant. Then for any $T > 1$, it holds that*

$$\mathbb{E}[F(\mathbf{w}_T)] - F(\mathbf{w}^*) \leq \left(\frac{D^2}{c} + cG^2\right)\frac{2 + \log(T)}{\sqrt{T}}.$$

Proof. The proof begins the same as in Thm. 1 (this time letting k be an element in $\{1, \ldots, T-1\}$), up to Eq. (1). Instead of substituting $\eta_t = c/\lambda t$, we substitute $\eta_t = c/\sqrt{t}$, to get the, $\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}\|^2]$ by D^2, pick $\mathbf{w} = \mathbf{w}_{T-k}$ and slightly simplify to get

$$\mathbb{E}\left[\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{T-k}\rangle\right]$$

$$\leq \frac{D^2}{2c}\left(\sqrt{T} - \sqrt{T-k}\right) + \frac{G^2}{2}\sum_{t=T-k}^{T}\frac{c}{\sqrt{t}}.$$

By convexity, we can lower bound $\langle\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{T-k}\rangle$ by $F(\mathbf{w}_t) - F(\mathbf{w}_{T-k})$. Also, it is easy to verify (e.g. by integration) that $\sum_{t=T-k}^{T}\frac{1}{\sqrt{t}} \leq 2(\sqrt{T} - \sqrt{T-k-1})$, hence

$$\mathbb{E}\left[\sum_{t=T-k}^{T}(F(\mathbf{w}_t) - F(\mathbf{w}_{T-k}))\right]$$

$$\leq \left(\frac{D^2}{2c} + cG^2\right)\left(\sqrt{T} - \sqrt{T-k-1}\right)$$

$$= \left(\frac{D^2}{2c} + cG^2\right)\frac{k+1}{\sqrt{T} + \sqrt{T-k-1}}$$

$$\leq \left(\frac{D^2}{2c} + cG^2\right)\frac{k+1}{\sqrt{T}}. \quad (5)$$

As in the proof of Thm. 1, let $S_k = \frac{1}{k+1}\sum_{t=T-k}^{T}\mathbb{E}[F(\mathbf{w}_t)]$ be the expected average value of the last $K+1$ iterates. The bound above implies that

$$-\mathbb{E}[F(\mathbf{w}_{T-k})] \leq -\mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{\sqrt{T}}.$$

By the definition of S_k and the inequality above, we have

$$k\mathbb{E}[S_{k-1}] = (k+1)\mathbb{E}[S_k] - \mathbb{E}[F(\mathbf{w}_{T-k})]$$

$$\leq (k+1)\mathbb{E}[S_k] - \mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{\sqrt{T}},$$

and dividing by k, implies

$$\mathbb{E}[S_{k-1}] \leq \mathbb{E}[S_k] + \frac{D^2/2c + cG^2}{k\sqrt{T}}.$$

Using this inequality repeatedly and by summing over $k = 1, \ldots, T-1$, we have

$$\mathbb{E}[F(\mathbf{w}_T)] = \mathbb{E}[S_0] \leq \mathbb{E}[S_{T-1}] + \frac{D^2/2c + cG^2}{\sqrt{T}}\sum_{k=1}^{T-1}\frac{1}{k}. \quad (6)$$

It now just remains to bound the terms on the right hand side. Using Eq. (1) with $k = T-1$ and $\mathbf{w} = \mathbf{w}^*$,

and upper bounding the norms by D, it is easy to calculate that

$$\mathbb{E}[S_{T-1}] - F(\mathbf{w}^*) = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)]\right]$$

$$\leq \left(\frac{D^2}{c} + cG^2\right)\frac{1}{\sqrt{T}}.$$

Also, we have $\sum_{k=1}^{T-1} 1/k \leq (1 + \log(T))$. Plugging these upper bounds into Eq. (6) and simplifying for readability, we get the required bound. $\qquad\square$

4. Averaging Schemes

The bounds shown in the previous section imply that individual iterates $\mathbf{w}_T$ have $\mathcal{O}(\log(T)/T)$ expected error in the strongly convex case, and $\mathcal{O}(\log(T)/\sqrt{T})$ expected error in the convex case. These bounds are close but not the same as the minimax optimal rates, which are $\mathcal{O}(1/T)$ and $\mathcal{O}(1/\sqrt{T})$ respectively. In this section, we consider averaging schemes, which rather than return individual iterates, return some weighted combination of all iterates $\mathbf{w}_1, \ldots, \mathbf{w}_T$, attaining the minimax optimal rates. We mainly focus here on the strongly-convex case, since simple averaging of all iterates is already known to be optimal (up to constants) in the general convex case.

We first examine the case of *α-suffix averaging*, defined as the average of the last αT iterates (where $\alpha \in (0, 1)$ is a constant, and αT is assumed to be an integer):

$$\bar{\mathbf{w}}_T^{\alpha} = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbf{w}_t.$$

In (Rakhlin et al., 2011), it was shown that this averaging scheme results in an optimization error of $\mathcal{O}((1 + \log(\frac{1}{1-\alpha}))/\alpha T)$, which is optimal in terms of T, but increases rapidly as we make α smaller. The following theorem shows a tighter upper bound of $\mathcal{O}(\log(\frac{1}{\min\{\alpha, 1-\alpha\}})/T)$, which implies we can be much more flexible in choosing α. Besides being of independent interest, we will re-use this result in our proofs later on.

Theorem 3. *Under the conditions of Thm. 1, and assuming αT is an integer, it holds that $\mathbb{E}[F(\bar{\mathbf{w}}_T^{\alpha}) - F(\mathbf{w}^*)]$ is at most*

$$\frac{17G^2\left(1 + \log\left(\frac{1}{\min\{\alpha, (1+1/T)-\alpha\}}\right)\right)}{\lambda T}.$$

Proof. Suppose first that $\alpha T \leq \lfloor T/2 \rfloor$. The proof is mostly identical to that of Thm. 1, except that instead

of using Eq. (3) to bound $\mathbb{E}[S_0]$, we use it to bound $\mathbb{E}[S_{\alpha T-1}] = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T} F(\mathbf{w}_t)$, which by convexity upper bounds $F(\bar{\mathbf{w}}_T^{\alpha})$. We get:

$$\mathbb{E}[S_{\alpha T-1}] \leq \mathbb{E}[S_{\lfloor T/2 \rfloor}] + \frac{16G^2}{\lambda T}\sum_{k=\alpha T}^{\lfloor T/2 \rfloor}\frac{1}{k}$$

$$+ \frac{G^2}{2\lambda}\sum_{k=\alpha T}^{\lfloor T/2 \rfloor}\sum_{t=T-k}^{T}\frac{1}{k(k+1)t},$$

Using the same argument as in the proof of Thm. 1, and the fact that $\sum_{k=\alpha T}^{\beta T}\frac{1}{k} \leq 1 + \log(\beta/\alpha)$ for any integers $\alpha T, \beta T$ that are no larger than T, we can obtain the upper bounds

$$\mathbb{E}[S_{\lfloor T/2 \rfloor}] \leq F(\mathbf{w}^*) + 10G^2/\lambda T$$

$$\sum_{k=\alpha T}^{\lfloor T/2 \rfloor}\frac{1}{k} \leq 1 + \log(1/2\alpha)$$

and

$$\sum_{k=\alpha T}^{\lfloor T/2 \rfloor}\sum_{t=T-k}^{T}\frac{1}{k(k+1)t} \leq \frac{1}{T}\sum_{k=\alpha T}^{\lfloor T/2 \rfloor}\left(\frac{1}{k} + \frac{1}{T-k}\right)$$

$$\leq \frac{1}{T}\left((1 + \log(1/2\alpha)) + (1 + \log(2(1-\alpha)))\right)$$

$$\leq \frac{1}{T}\left(2 + \log(1/\alpha)\right).$$

Using the above estimates, with some simplifications for readability, we get that $\mathbb{E}[F(\bar{\mathbf{w}}_T^{\alpha}) - F(\mathbf{w}^*)]$ is at most

$$17\left(1 + \log\left(\frac{1}{\alpha}\right)\right)\frac{G^2}{\lambda T}. \qquad (7)$$

This analysis assumed $\alpha T \leq \lfloor T/2 \rfloor$. If α is larger, we can use the existing analysis ((Rakhlin et al., 2011), Theorem 5), and get that $\mathbb{E}[F(\bar{\mathbf{w}}_T^{\alpha}) - F(\mathbf{w}^*)]$ is at most

$$\left(4 + 5\log\left(\frac{1}{1 + 1/T - \alpha}\right)\right)\frac{G^2}{\lambda T}. \qquad (8)$$

Combining Eq. (7) and Eq. (8) with a uniform upper bound which holds for all α, we get the required bound. $\qquad\square$

We note that in the general convex case without assuming strong convexity, one can use an analogous proof to show an upper bound of order $\log(1/\alpha)/\sqrt{T}$ for α-suffix averaging. In contrast, existing techniques only imply a bound of order $1/\sqrt{\alpha T}$.

As discussed in the introduction, a limitation of suffix averaging is that unless we can store all iterates in memory, it requires us to guess the stopping time T

in advance. For example, if we do 1/2-suffix averaging, we need to "know" when we got to iterate $T/2$ and should start averaging. In practice, the stopping time T is often not known in advance and is determined empirically (e.g. till satisfactory performance is obtained). One way to handle this is to decide in advance on a fixed schedule of stopping times T (e.g. $T_0, 2T_0, 2^2 T_0, 2^3 T_0, \ldots$ for some T_0) and maintain suffix-averages only for those times. However, this is still not very flexible. In contrast, maintaining the average of all iterates up to time t can be done on-the-fly: we initialize $\bar{\mathbf{w}}_1 = \mathbf{w}_1$, and for any $t > 1$, we let

$$\bar{\mathbf{w}}_t = \left(1 - \frac{1}{t}\right)\bar{\mathbf{w}}_{t-1} + \frac{1}{t}\mathbf{w}_t. \tag{9}$$

Unfortunately, returning the average of all iterates as in Eq. (9) is provably suboptimal and can harm performance (Rakhlin et al., 2011). Alternatively, we can easily maintain and return the current iterate $\mathbf{w}_t$, but we only have a suboptimal $\mathcal{O}(\log(t)/t)$ bound for it.

In the following, we analyze a new and very simple running average scheme, denoted as *polynomial-decay averaging*, and show that it combines the best of both worlds: it can easily be computed on the fly, and it gives an optimal rate. It is parameterized by a number $\eta \geq 0$, which should be thought of as a small constant (e.g. $\eta = 3$), and the procedure is defined as follows: $\bar{\mathbf{w}}_1^\eta = \mathbf{w}_1$, and for any $t > 1$,

$$\bar{\mathbf{w}}_t^\eta = \left(1 - \frac{\eta+1}{t+\eta}\right)\bar{\mathbf{w}}_{t-1}^\eta + \frac{\eta+1}{t+\eta}\mathbf{w}_t. \tag{10}$$

For $\eta = 0$, this is exactly standard averaging (see Eq. (9)), whereas $\eta > 0$ reduces the weight of earlier iterates compared to later ones. Moreover, $\bar{\mathbf{w}}_t^\eta$ can be computed on-the-fly, just as easily as computing a standard average.

We note that after this paper was accepted for publication, a similar averaging scheme was independently proposed and studied in (Lacoste-Julien et al., 2012). Compared to our method, they consider a slightly different step-size and a specific choice of $\eta = 1$, using a more direct proof technique tailored to this case.

An analysis of our averaging scheme is provided in the theorem below.

Theorem 4. *Suppose F is λ-strongly convex, and that $\mathbb{E}[\|\hat{\mathbf{g}}_t\|^2] \leq G^2$ for all t. Consider SGD initialized with $\mathbf{w}_1$ and step-sizes $\eta_t = 1/\lambda t$. Also, let $\eta \geq 1$ be an integer. Then $\mathbb{E}[F(\mathbf{w}_T^\eta) - F(\mathbf{w}^*)]$ is at most*

$$58\left(1 + \frac{\eta}{T}\right)\left(\eta(\eta+1) + \frac{(\eta+0.5)^3(1+\log(T))}{T}\right)\frac{G^2}{\lambda T}.$$

The assumption that η is an integer is merely for simplicity. Also, we made no effort to optimize the constants, which can be easily improved for specific choices of η (see the proof as well as the analysis in (Lacoste-Julien et al., 2012)).

Proof. We can rewrite the recursion as

$$\bar{\mathbf{w}}_t^\eta = \frac{t-1}{t+\eta}\bar{\mathbf{w}}_{t-1}^\eta + \frac{\eta+1}{t+\eta}\mathbf{w}_t$$

for $t \geq 1$ with $\bar{\mathbf{w}}_0^\eta = 0$. Unwrapping the recursion, we have that for any $T \geq 1$, $\bar{\mathbf{w}}_T^\eta = \sum_{t=1}^T \alpha_t \mathbf{w}_t,$, where

$$\alpha_t = \frac{\eta+1}{t+\eta}\prod_{j=t+1}^T \frac{j-1}{j+\eta},$$

and at $t = T$, the convention that $\prod_{j=T+1}^T ((j-1)/(j+\eta)) = 1$ is used.

We now denote $F'(\mathbf{w}) = F(\mathbf{w}) - F(\mathbf{w}^*)$. Since $\bar{\mathbf{w}}_T^\eta$ is a weighted average of $\mathbf{w}_1, \ldots, \mathbf{w}_T$, where the weights α_t sum up to be 1, it follows by the convexity of F and Jensen's inequality that $F'(\bar{\mathbf{w}}_T^\eta) \leq \sum_{t=1}^T \alpha_t F'(\mathbf{w}_t)$.

Let $S_k' = \sum_{t=T-k}^T F'(\mathbf{w}_t)$, and $\alpha_0 = 0$, then we have

$$F'(\bar{\mathbf{w}}_T^\eta) \leq \sum_{t=1}^T (\alpha_t - \alpha_{t-1})S_{T-t}'. \tag{11}$$

It is not difficult to check that for all $t \geq 1$:

$$\alpha_t - \alpha_{t-1} = \frac{\eta(\eta+1)}{(t-1+\eta)(t+\eta)}\prod_{j=t+1}^T \frac{j-1}{j+\eta}$$

$$= \frac{\eta(\eta+1)}{T(T+1)}\prod_{j=t}^{T+1}\frac{j}{j-1+\eta}$$

$$\leq \begin{cases} \frac{\eta(\eta+1)}{T(T+1)}\left(\frac{t-2+\eta}{T+\eta}\right)^{\eta-1} & \text{if } t \leq T+2-\eta \\ \frac{\eta(\eta+1)}{T(T+1)} & \text{otherwise} \end{cases}. \tag{12}$$

Let us suppose first that $\eta \geq 2$. In that case, we can upper bound the above by

$$\frac{\eta(\eta+1)(t+\eta)}{T(T+1)(T+2)}.$$

As to S_{T-t}' in Eq. (11), note that the upper bound proof of Thm. 3 equally applies to $\frac{1}{T-t+1}S_{T-t}'$. Using

this bound and substituting in Eq. (11), we obtain

$F'(\bar{\mathbf{w}}_T^\eta)$

$$\leq \sum_{t=1}^{T}(\alpha_t - \alpha_{t-1})(T-t+1)\frac{17G^2\log\left(\frac{Te}{\min\{t,T-t+1\}}\right)}{\lambda T}$$

$$\leq \sum_{t=1}^{\lceil T/2\rceil}\frac{2\eta(\eta+1)(t+\eta)}{T(T+1)(T+2)}(T+\eta)\frac{17G^2\log(Te/t)}{\lambda T}$$

$$\leq \frac{34G^2\eta(\eta+1)(T+\eta)}{\lambda T^2(T+1)(T+2)}(A+B+C), \tag{13}$$

where

$$A = \sum_{t=1}^{\lceil T/2\rceil}\eta\log(Te/t) \leq \eta\frac{T+1}{2}\log(Te),$$

and

$$B = \sum_{t=1}^{\lceil T/2\rceil}t\log(Te) \leq 0.5(\lceil T/2\rceil)(\lceil T/2\rceil+1)\log(Te),$$

and

$$C \leq -\sum_{t=1}^{\lceil T/2\rceil}t\log(t) \leq -\int_{t=1}^{\lceil T/2\rceil}t\log(t)dt$$

$$= -\left[0.5t^2\log t - 0.25t^2\right]\Big|_1^{\lceil T/2\rceil}$$

$$= -0.5\lceil T/2\rceil^2\log(T/2) + 0.25\lceil T/2\rceil^2 - 0.25.$$

Therefore we have

$$A+B+C$$

$$\leq(\eta+0.5)\frac{T+1}{2}\log(Te)$$

$$\quad + 0.5\frac{(T+1)^2}{4}\log(2e^{1.5}) - 0.25$$

$$\leq(\eta+0.5)\frac{T+1}{2}\log(Te) + 0.5(T+1)(T+2).$$

Plugging this estimate into Eq. (13) and simplifying, we obtain an upper bound on $\mathbb{E}\left[F(\mathbf{w}_T^\eta) - F(\mathbf{w}^*)\right]$ of the form

$$17\left(1+\frac{\eta}{T}\right)\left(\eta(\eta+1) + \frac{(\eta+0.5)^3(1+\log(T))}{T}\right)\frac{G^2}{\lambda T}. \tag{14}$$

It remains to treat the case $\eta = 1$. In that case, the upper bound on $\alpha_t - \alpha_{t-1}$ in Eq. (12) becomes

$$\alpha_t - \alpha_{t-1} \leq \frac{2}{T(T+1)},$$

and using the same derivation as before, we get that

$$F'(\bar{\mathbf{w}}_T^\eta) \leq \sum_{t=1}^{\lceil T/2\rceil}\frac{68G^2\log(Te/t)}{\lambda T^2}.$$

By an integration calculation, it is easy to verify that

$$\sum_{t=1}^{\lceil T/2\rceil}\log(Te/t) \leq \left\lceil\frac{T}{2}\right\rceil\log(Te) - \int_{t=1}^{\lceil T/2\rceil}\log(t)dt$$

$$= \left\lceil\frac{T}{2}\right\rceil\log(Te) - [t\log(t) - t]\Big|_1^{\lceil T/2\rceil}$$

$$\leq \left\lceil\frac{T}{2}\right\rceil(2+\log(2)) - 1.$$

Plugging it in, we get an upper bound of

$$\frac{68G^2}{\lambda T}\frac{\lceil T/2\rceil(2+\log(2)) - 1}{T},$$

which for any $T \geq 1$ is at most $116G^2/\lambda T$. The stated result follows by combining this bound (for $\eta = 1$) and Eq. (14) (for $\eta \geq 2$), increasing the numerical constant in Eq. (14) to obtain a uniform bound which holds for all choices of η. $\qquad\square$

Note that for a constant η, the bound is essentially optimal. We end by noting that using an identical proof technique, it holds in the case of general convex F (with assumptions similar to Thm. 2) that

$$\mathbb{E}\left[F(\mathbf{w}_T^\eta) - F(\mathbf{w}^*)\right] \leq \mathcal{O}\left(\frac{\eta(D^2/c + cG^2)}{\sqrt{T}}\right),$$

this implies that polynomial-decay averaging is also optimal (up to constants) in the general convex case.

5. Experiments

In this section, we study the behavior of the polynomial-decay averaging scheme on a few strongly-convex optimization problems. We chose the same 3 binary classification datasets ((CCAT,COV1 and ASTRO-PH) and experimental setup as in (Rakhlin et al., 2011). For each dataset $\{\mathbf{x}_i, y_i\}_{i=1}^m$, we ran SGD on the support vector machine optimization problem

$$F(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\max\{0, 1 - y_i\langle\mathbf{x}_i, \mathbf{w}\rangle\},$$

with the domain $\mathcal{W} = \mathbb{R}^d$, where the stochastic gradient given $\mathbf{w}_t$ was computed by taking a single randomly drawn training example $(\mathbf{x}_i, y_i)$ and computing the gradient with respect to that example, i.e. $\hat{\mathbf{g}}_t = \lambda\mathbf{w}_t - \mathbf{1}_{y_i\langle\mathbf{x}_i,\mathbf{w}_t\rangle\leq 1}y_i\mathbf{x}_i$. All algorithms were initialized at $\mathbf{w}_1 = 0$. Following previous work, we chose $\lambda = 10^{-4}$ for CCAT, $\lambda = 10^{-6}$ for COV1, and $\lambda = 5 \times 10^{-5}$ for ASTRO-PH. The η parameter of polynomial-decay averaging was set to 3. For comparison, besides polynomial-decay averaging, we also

ran suffix averaging with $\alpha = 1/2$, and simple averaging of all iterates. The results are reported in the figure below. Each graph is a log-log plot representing the training error on one dataset over 10 repetitions, as a function of the number of iterations. We also experimented on the test set provided with each dataset, but omit the results as they are very similar.

The graphs below clearly indicate that polynomial-decay averaging work quite well. Achieving the best or almost-best performance in all cases. Suffix averaging performs performs similarly, although as discussed earlier, it is not as amenable to on-the-fly computation. Compared to these schemes, a simple average of all iterates is significantly suboptimal, matching the results of (Rakhlin et al., 2011).

6. Discussion

In this paper, we investigated the convergence behavior of SGD, and the averaging schemes required to obtain optimal performance. In particular, we considered polynomial-decay averaging, which is as simple to compute as standard averaging of all iterates, but attains better performance theoretically and in practice. We also extended the existing analysis of SGD by providing new finite-sample bounds on individual SGD iterates, which hold without any smoothness assumptions, for both convex and strongly-convex problems. Finally, we provided new bounds for suffix averaging. While we focused on standard gradient descent, our techniques can be extended to the more general mirror descent framework and non-Euclidean norms.

An important open question is whether the $\mathcal{O}(\log(T)/T)$ rate we obtained on the individual iterate $\mathbf{w}_T$, for strongly-convex problems, is tight. This question is important, because running SGD for T iterations, and returning the last iterate $\mathbf{w}_T$, is a very common heuristic. If the $\mathcal{O}(\log(T)/T)$ bound is tight, it means practitioners should *not* return the last iterate, since better $\mathcal{O}(1/T)$ rates can be obtained by suffix averaging or polynomial-decay averaging. Alternatively, a $\mathcal{O}(1/T)$ bound on the last iterate can indicate that returning the last iterate is indeed justified. For a further discussion of this, see (Shamir, 2012). Another question is whether high-probability versions of our individual iterate bounds (Thm. 1 and Thm. 2) can be obtained, especially in the strongly-convex case. Again, this question has practical implications, since if a high-probability bound does not hold, it might imply that the last iterate can suffer from high variability, and should be used with caution. Finally, the tightness of Thm. 2 is still unclear. In fact, even for the simpler case of
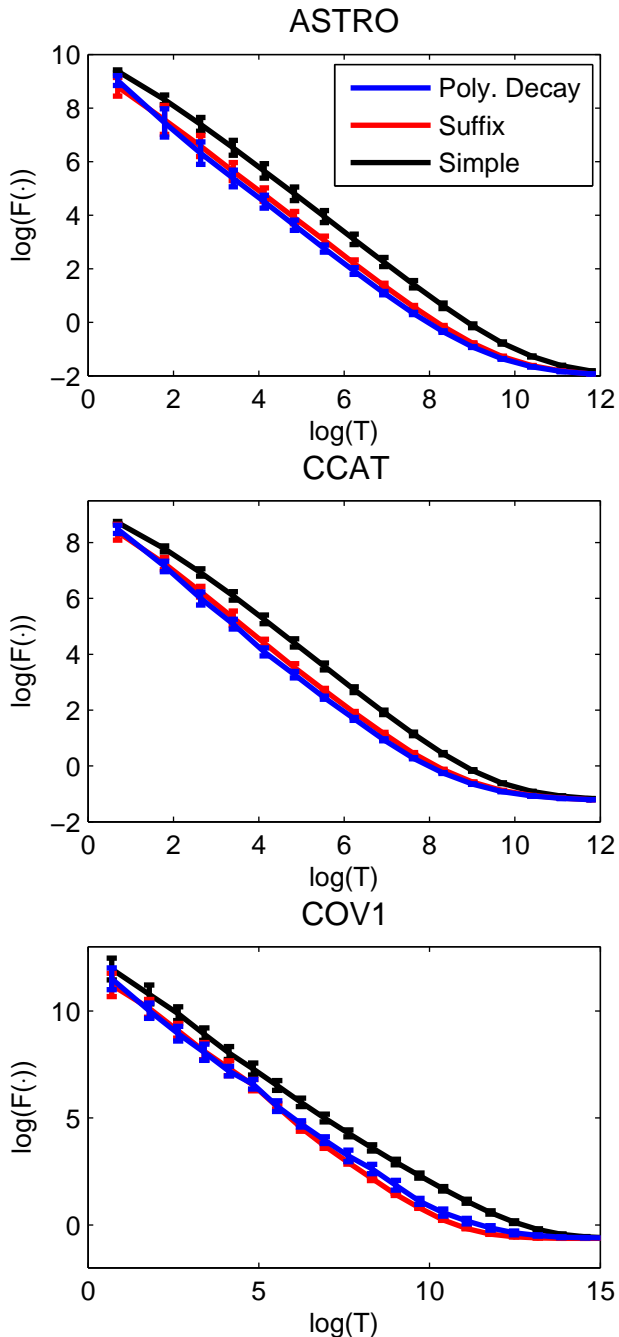
(non-stocahstic) gradient descent, we do not know whether the behavior of the last iterate proved in Thm. 2 is tight. In general, for an algorithm as simple and popular as SGD, we should have a better understanding of how it behaves and how it should be used in an optimal way.

References

Agarwal, A., Bartlett, P., Ravikumar, P., and Wainwright, M. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58 (5):3235–3249, 2012.

Bach, F. and Moulines, E. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *NIPS*, 2011.

Hazan, E. and Kale, S. Beyond the regret minimization barrier: An optimal algorithm for stochastic strongly-convex optimization. In *COLT*, 2011.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Kushner, H. and Yin, G. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2nd edition, 2003.

Lacoste-Julien, S., Schmidt, M., and Bach, F. A simpler approach to obtaining an o(1/t) convergence rate for projected stochastic subgradient descent. *CoRR*, abs/1212.2002, 2012.

Ouyang, H. and Gray, A. Stochastic smoothing for nonsmooth minimizations: Accelerating sgd by exploiting structure. In *ICML*, 2012.

Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. *CoRR*, abs/1109.5647, 2011.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *COLT*, 2009.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. Pegasos: primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

Shamir, O. Is averaging needed for strongly convex stochastic gradient descent? Open problem presented at COLT, 2012.

Zhang, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.