SIAM J. OPTIM.
Vol. 30, No. 2, pp. 1473–1500

ON THE ADAPTIVITY OF STOCHASTIC GRADIENT-BASED OPTIMIZATION*

LIHUA LEI† AND MICHAEL I. JORDAN‡

Abstract. Stochastic gradient-based optimization has been a core enabling methodology in applications to large-scale problems in machine learning and related areas. Despite this progress, the gap between theory and practice remains significant, with theoreticians pursuing mathematical optimality at the cost of obtaining specialized procedures in different regimes (e.g., modulus of strong convexity, magnitude of target accuracy, signal-to-noise ratio), and with practitioners not readily able to know which regime is appropriate to their problem, and seeking broadly applicable algorithms that are reasonably close to optimality. To bridge these perspectives it is necessary to study algorithms that are *adaptive* to different regimes. We present the stochastically controlled stochastic gradient (SCSG) method for composite convex finite-sum optimization problems and show that it is adaptive to both strong convexity and target accuracy. The adaptivity is achieved by batch variance reduction with adaptive batch sizes and a novel technique, which we refer to as *geometrization*, and which sets the length of each epoch as a geometric random variable. The algorithm achieves strictly better theoretical complexity than other existing adaptive algorithms, while the tuning parameters of the algorithm depend only on the smoothness parameter of the objective.

Key words. adaptivity, stochastic gradient method, finite-sum optimization, geometrization, variance reduction

AMS subject classifications. 90C15, 90C25, 90C06

DOI. 10.1137/19M1256919

1. Introduction.

The application of gradient-based optimization methodology to statistical machine learning has been a major success story in both practice and theory. Indeed, there is an increasingly detailed theory available for gradient-based algorithms that helps to explain their practical success. There remains, however, a significant gap between theory and practice, in that the designer of machine learning algorithms is required to make numerous choices that depend on parameters that are unlikely to be known in a real-world machine-learning setting. For example, existing theory asserts that different algorithms are preferred if a problem is strongly convex or merely convex, if the target accuracy is high or low, if the signal-to-noise is high or low, and if data are independent or correlated. This poses a serious challenge to builders of machine-learning software as well as users of that software. Indeed, a distinctive aspect of machine-learning problems, especially large-scale problems, is that the user of an algorithm can be expected to know little or nothing about the quantitative structural properties of the functions being optimized. It is hoped that the data and the data analysis will inform such properties, not the other way around.

A classical example is the stochastic gradient descent (SGD) algorithm, which takes different forms for strongly convex objectives and non–strongly convex objectives. In the former case, letting μ denote the strong-convexity parameter, if the stepsize is set as $O(1/\mu t)$, then the SGD exhibits a convergence rate of $O(1/\mu\epsilon)$, where ϵ is the target accuracy [35]. In the latter case, setting the stepsize to $O(1/\sqrt{t})$

*Received by the editors April 17, 2019; accepted for publication (in revised form) March 2, 2020; published electronically May 27, 2020. This work was completed while the first author was a graduate student in the Department of Statistics at UC Berkeley.

https://doi.org/10.1137/19M1256919

†Department of Statistics, Stanford University, Stanford, CA 94305 (lihualei@stanford.edu).

‡Department of Statistics and Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720 (jordan@cs.berkeley.edu).

yields a rate of $O(1/\epsilon^2)$ [34]. Using the former scheme for non–strongly convex objectives can significantly deteriorate the convergence [34]. It is sometimes suggested that one can ensure strong convexity by simply adding a quadratic regularizer to the objective, using the coefficient of the regularizer as a conservative estimate of the strong-convexity parameter. But this produces a significantly faster rate only if $\mu \gg \epsilon$, a regime that is unrealistic in many machine-learning applications, where ϵ is relatively large. Setting μ to such a large value would have a major effect on the statistical properties of the optimizer.

Similar comments apply to presumptions of knowledge of Lipschitz parameters, mini-batch sizes, variance-reduction tuning parameters, etc. Current practice often involves heuristics in setting these tuning parameters, but the use of these heuristics can change the algorithm, and the optimality guarantees may disappear.

Our goal, therefore, should be that our algorithms are *adaptive*, in the sense that they perform as well as an algorithm that is assumed to know the "correct" choice of tuning parameters, even if they do not know those parameters. In particular, in the convex setting, we wish to derive an algorithm that does not involve μ in its implementation but whose convergence rate would be better for larger μ while still reasonable for smaller μ, including the non–strongly convex case where $\mu = 0$.

Such adaptivity has been studied implicitly in the classic literature. The authors of [45, 41, 42] showed that the average iterate of SGD with stepsize $O(t^{-\alpha})$ for $\alpha \in (1/2, 1)$ satisfies a central limit theorem with information-theoretic optimal asymptotic variance. This implies adaptivity because the performance adapts to the underlying parameters of the problem, including the modulus of strong convexity, even though the algorithm does not require knowledge of them. The analysis by [42] is, however, asymptotic and relies on the smoothness of the Hessian. Under similar assumptions on the Hessian, [33] provided a nonasymptotic analysis establishing adaptivity of SGD with Polyak–Ruppert averaging. Further contributions to this line of work include [7, 15, 11], which prove the adaptivity of certain versions of SGD with refined rates for self-concordant objectives, including least-square regression and logistic regression.

This line of work relies on conditions on higher-order derivatives, which are not required in the modern literature on stochastic gradient methods. In fact, under fairly standard assumptions for first-order methods, Moulines and Bach [33] provided a non-asymptotic analysis for SGD with stepsize $O(t^{-\alpha})$ without averaging and showed that this algorithm exhibits adaptivity to strong convexity while having a reasonable guarantee for non–strongly convex objectives. Specifically, if $\alpha = 2/3$, their results show that the rate for achieving an ϵ-accurate solution for the expected function value is $\tilde{O}\big(\min\{1/\mu^3 + 1/\mu\epsilon^2,\ 1/\epsilon^3\}\big)$, where $\tilde{O}$ hides logarithmic factors. This result was taken further by studying alternative stepsize schemes; in particular, [54] proposed a variant of projected SGD with stagewise diminishing stepsizes and diameters. Unlike the aforementioned work, the adaptivity in this case is weaker because it requires knowledge of the strong convexity parameter, as well as the initial suboptimality, in the complexity analysis (in particular, they require a sufficiently large initial diameter). A weaker form of adaptivity, to smoothness but not to strong convexity, was established by [28] when the polynomial decaying stepsize is replaced by an AdaGrad-type stepsize [12], assuming a bounded domain and bounded stochastic gradients. Finally, [9] presented a restarting variant of AdaGrad with provable adaptivity to strong convexity given an initial overestimate of the strong convexity parameter.

Further progress has been made by focusing on a setting that is particularly relevant to machine learning—that of *finite-sum optimization*. The objective function

in this setting takes the following form:

$$\min_{x\in\mathcal{X}} F(x) = f(x) + \psi(x), \quad \text{where } f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1.1}$$

where $\mathcal{X}$ is the parameter space, n is the number of data points, the functions $f_i(x)$ are data-point–specific *loss* functions, and $\psi(x)$ is the *regularization* term. We assume that each $f_i(x)$ is differentiably convex and that $\psi(x)$ is convex but can be nondifferentiable. The introduction of the parameter n into the optimization problem has two important implications. First, it implies that the number of operations for obtaining a full gradient is $O(n)$, which is generally impractical in modern machine-learning applications, where the value of n can be in the tens to hundreds of millions. This fact motivates us to make use of stochastic estimates of gradients. Such randomness introduces additional variance that interacts with the variability of the data, and tuning parameters are often introduced to control this variance.

Second, the finite-sum formulation highlights the need for adaptivity to the target accuracy ϵ, where that accuracy is related to the number of data points n for statistical reasons. Unfortunately, different algorithms perform better in high-accuracy versus low-accuracy regimes, and the choice of regime is generally not clear to a user of machine-learning algorithms, given that target accuracy varies not only as a function of n but also as a function of other parameters, such as the signal-to-noise ratio, that the user is not likely to know. Ideally, therefore, optimization algorithms should be adaptive to target accuracy, performing well in either regime.

Deterministic gradient-descent–based methods can be made adaptive to strong convexity and smoothness simultaneously by exploiting the Polyak stepsize [17]. However, computation of a full gradient is expensive, rendering the method undesirable for finite-sum optimization. A recent line of research has shown that algorithms with lower complexity can be designed in the finite-sum setting with some adaptivity, generally via careful control of the variance. The stochastic average gradient (SAG) method opened this line of research, establishing the complexity of $\tilde{O}\left(\min\{n/\epsilon, n + L/\mu\}\right)$ [44]. Importantly, this result shows that SAG is adaptive to strong convexity. To achieve such adaptivity, however, SAG requires two sequences of iterates: the average iterate and the last iterate. Defazio, Bach, and Lacoste-Julien [10] propose a single-sequence variant of SAG that is also adaptive to strong convexity, yet under the stronger assumption that each f_i is strongly convex. Both methods suffer, however, from a prohibitive storage cost of $O(nd)$, where d is the dimension of $\mathcal{X}$. Further developments in this vein include the stochastic variance reduced gradient (SVRG) method [19] and the stochastic dual coordinate ascent (SDCA) method [19]; they achieve the same computational complexity as SAG while reducing the storage cost to $O(d)$. They are not, however, adaptive to strong convexity.

Lei and Jordan [25] presented a randomized variant of SVRG that achieves the same convergence rate and adaptivity as SAG but with the same storage cost as SVRG. However, as is the case with SAG, the complexity of $O(n/\epsilon)$ for the non–strongly convex case is much larger than the oracle lower bound of $O(n + \sqrt{n/\epsilon})$ [52]. Nguyen et al. [37] proposed another variance-reduction method that is provably adaptive to strong convexity, though the result is proved for the expected gradient norm and is thus weaker than that of [25]. Xu, Lin, and Wang [53] developed another variant of SVRG which adapts to a more general condition, called a "Hölderian error bound," with strong convexity being a special case. In contrast to [25], they required an initial conservative estimate of the strong convexity parameter. Under an

extra strong assumption of gradient interpolation—that all individual loss functions have vanishing gradients at the optimum—[51] developed an algorithm that achieves adaptivity to the smoothness parameter and to the modulus of strong convexity simultaneously. On the other hand, recent work of [24] that appeared after our work presented an algorithm that achieves certain adaptivity to the target accuracy. However, they need to know the modulus of strong convexity, and the adaptivity is only obtained in the high-accuracy regime because it requires full gradient computations periodically. Finally, while our focus is convex optimization, we note that adaptivity has also been studied for nonconvex finite-sum optimization (e.g., [27, 39]).

In this article we present the stochastically controlled stochastic gradient (SCSG) algorithm, which exhibits adaptivity to both strong convexity and target accuracy. SCSG is a nested procedure that is similar to the SVRG algorithm. Crucially, it does not require the computation of a full gradient in the outer loop as performed by SVRG but makes use of stochastic estimates of gradients in both the outer loop and the inner loop. Moreover, it makes essential use of a randomization technique ("geometrization") that allows terms to telescope across the outer and inner loops; such telescoping does not happen in SVRG, a fact which leads to the loss of adaptivity for SVRG.

The rest of this article is organized as follows. Section 2 introduces notation, assumptions and definitions. In sections 3 and 4, we focus on the relatively simple setting of unregularized problems and Euclidean geometry, introducing the key ideas of geometrization and adaptive batching. We provide key proofs in section 4 and leave nonessential proofs to Appendix A in [26]. We extend these results to regularized problems and to non-Euclidean geometry in section 5. The extension relaxes standard assumptions for analyzing mirror-descent methods and may be of independent interest. All proofs for the general case are relegated to Appendix B, and some miscellaneous results are presented in Appendix C in [26]. Finally, the desirable empirical performance of SCSG is demonstrated in Appendix D.

2. Notation, assumptions, and definitions. We write $a \wedge b$ (resp., $a \vee b$) for $\min\{a, b\}$ (resp., $\max\{a, b\}$) and $(a)_*^\xi$ (or $[a]_*^\xi$) for $\max\{a, 1\}^\xi$ throughout the paper. The symbol $\mathbb{E}$ denotes the expectation of a random element, and $\mathbb{E}_X$ denotes an expectation over the randomness of X while conditioning on all other random elements. We adopt Landau's notation $(O(\cdot), o(\cdot))$, and we occasionally use $\tilde{O}(\cdot)$ to hide logarithmic factors. We define computational cost by making use of the IFO (incremental first-order oracle) framework of [1, 43], where we assume that sampling an index i and computing the pair $(f_i(x), \nabla f_i(x))$ incur a unit of cost. For notational convenience, given a subset $\mathcal{I} \subset \{1, \dots, n\}$, we denote by $\nabla f_{\mathcal{I}}(x)$ the batch gradient,

$$\nabla f_{\mathcal{I}}(x) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \nabla f_i(x).$$

By definition, computing $\nabla f_{\mathcal{I}}(x)$ incurs $|\mathcal{I}|$ units of cost.

In this section and sections 3 and 4, we focus on unregularized problems and Euclidean geometry, turning to regularized problems and non-Euclidean geometry in section 5. Specifically, we consider the case $\mathcal{X} = \mathbb{R}^d$, $\psi \equiv 0$ and make the following assumptions that target the finite-sum optimization problem:

A1. f_i is convex with L-Lipschitz gradient

$$f_i(x) - f_i(y) - \langle \nabla f_i(y), x - y \rangle \le \frac{L}{2} \|x - y\|_2^2 \quad \forall i = 1, \dots, n$$

for some $L < \infty$.

A2. $F = f$ is strongly convex at x^* with

$$f(x) - f(x^*) \geq \frac{\mu}{2}\|x - x^*\|_2^2$$

for some $\mu \geq 0$.

Note that assumption A2 always holds with $\mu = 0$, corresponding to the non–strongly convex case. Note also that with the exception of [44], this assumption is weaker than most of the literature on smooth finite-sum optimization, where strong convexity of f is required at every point.

Our analysis will make use of the following key quantity [25]:

$$\mathcal{H}(f) = \frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x^*)\|_2^2,$$

where x^* denotes the optimum of f. If multiple optima exist, we take one that minimizes $\mathcal{H}(f)$. We use $\mathcal{H}(f)$, an average squared norm at the optimum, in place of the uniform upper bound on the gradient that is often assumed in other work. The latter is not realistic for many practical problems in machine learning, including least squares, where the gradient is unbounded. On the other hand, it is noteworthy that $\mathcal{H}(f)$ depends only on the optimum. For instance, $\mathcal{H}(f) = 0$ if all gradients vanish at the optimum, as studied for overparametrized models (e.g., [32, 50, 51]). As will be shown later, the complexity of our algorithm depends only on $\mathcal{H}(f)$, so it can be applied to studying the case with data interpolation. We will write $\mathcal{H}(f)$ as $\mathcal{H}$ when no confusion can arise.

We let $\tilde{x}_0$ denote the initial value (possibly random) and define the following measures of complexity:

$$D_x = L\mathbb{E}\|\tilde{x}_0 - x^*\|_2^2, \quad D_H = \mathcal{H}/L, \quad D = \max\{D_x, D_H\}. \tag{2.1}$$

Recall that a geometric random variable, $N \sim \text{Geom}(\gamma)$, is a discrete random variable with probability mass function $P(N = k) = (1-\gamma)\gamma^k$, for $k = 0, 1, \ldots$, and expectation

$$\mathbb{E}N = \frac{\gamma}{1-\gamma}.$$

Geometric random variables will play a key role in the design and analysis of our algorithm.

Finally, we introduce two fundamental definitions that serve to clarify desirable properties of optimization algorithms. We refer to the first property as ϵ-*independence*.

DEFINITION 2.1. *An algorithm is* ϵ-independent *if it guarantees convergence at all target accuracies* ϵ.

ϵ-independence is a crucial property in practice because a target accuracy is usually not exactly known a priori. An ϵ-independent algorithm satisfies the "one-pass-for-all" property where the theoretical complexity analysis applies to the whole path of the iterates. In contrast, an ϵ-dependent algorithm only has a theoretical guarantee for a particular ϵ, whose value is often unknown in practice. To illustrate we consider SGD, where the iterate is updated by $x_{k+1} = x_k - \eta_k \nabla f_{i_k}(x_k)$ and where i_k is a uniform index from $\{1, \ldots, n\}$. There are two popular schemes for theoretical analysis: (1) $\eta_k = O(1/\sqrt{k})$, or (2) $\eta_k \equiv 1/\sqrt{T}$ and the iterates are updated for $O(T)$ steps

where $T = O(1/\epsilon^2)$. Although both versions have theoretical complexity $\tilde{O}\left(\epsilon^{-2}\right)$, only the former is ϵ-independent.

The second important property is referred to as *almost universality*.

DEFINITION 2.2. *An algorithm is* almost universal *if it only requires the knowledge of the smoothness parameters L.*

The term *almost universality* is motivated by the notion of *universality* introduced by [36] which does not require the knowledge of L or other parameters, such as the variance of the stochastic gradients. Returning to the previous example, we note that both versions of SGD are universal. It is noteworthy that universal gradient methods are usually either ϵ-dependent (e.g., [36]) or require imposing other assumptions such as uniformly bounded ∇f_i (e.g., [34]). The SCSG algorithm developed in this paper is both ϵ-independent and almost universal. This category also includes SGD for general convex functions [34], SAG (Stochastic Average Gradient) [44], SAGA [10], SVRG++ [5], Katyusha for non–strongly convex functions [3], and AMSVRG (Accelerated efficient Mini-batch SVRG) [38]. In contrast, algorithms such as SGD for strongly convex functions [34], SVRG [19], SDCA [46], APCG [31], Katyusha for strongly convex functions [3], and adaptive SVRG [53] are ϵ-independent but not almost universal because they need full or partial knowledge of μ. Furthermore, algorithms such as Catalyst [30] and AdaptReg [4] even depend on unknown quantities, such as $F(x_0) - F(x^*)$ or the variance of the ∇f_i. In comparing algorithms, we believe that clarity on these distinctions, in addition to comparison of convergence rates, is critical.

3. Stochastically controlled stochastic gradient (SCSG). In this section we present SCSG, a computationally efficient framework for variance reduction in SGD algorithms. SCSG builds on the SVRG algorithm of [19], incorporating several essential modifications that yield not only computational efficiency but also adaptivity. Recall that SVRG is a nested procedure that computes a full gradient in each outer loop and uses that gradient as a baseline to reduce the variance of the stochastic gradients that are computed in an inner loop. The need to compute a full gradient, at a cost of n operations, unfortunately makes the SVRG procedure impractical for large-scale applications. SCSG seeks to remove this bottleneck by replacing the full gradient with an approximate, stochastic gradient, one that is based on a batch size that is significantly smaller than n but larger than the size used for the stochastic gradients in the inner loop. By carefully weighing the contributions to the bias and variance of these sampling-based estimates, SCSG achieves a small iteration complexity while also keeping the per-iteration complexity feasibly small.

Further support for the SCSG framework comes from the comparison with SVRG in the setting of strongly convex objectives. In this setting, SVRG relies heavily on a presumption of knowledge of the strong convexity parameter μ. In particular, to achieve a complexity of $O((n+\kappa)\log(1/\epsilon))$, the number of stochastic gradients queried in the inner loop of SVRG needs to scale as κ. By contrast, the SCSG framework achieves the same complexity without knowledge of μ. This is achieved by setting the number of inner-loop stochastic gradients to be a geometric random variable. As we discuss below, the usage of a geometric random variable—a technique that we refer to as "geometrization"—is crucial in the design and analysis of SCSG. We believe that it is a key theoretical tool for achieving adaptivity to strong convexity.

The original version of SCSG was ϵ-dependent and not almost universal, because it required knowledge of the parameter $\mathcal{H}$ [25]. Moreover the algorithm had a suboptimal rate in the high-accuracy regime. In further development of the SCSG framework, in

the context of nonconvex optimization [27], we found that ϵ-independence and almost universality could be achieved by employing an increasing sequence of batch sizes.

In the remainder of this section, we bring these ideas together and present the general form of the SCSG algorithm, incorporating adaptive batching, geometrization, and mini-batches into the inner loop. The resulting algorithm is adaptive, ϵ-independent, and almost universal. Roughly speaking, the adaptive batching enables the adaptivity to target accuracy, and the geometrization enables the adaptivity to strong convexity. The pseudocode for SCSG is shown in Algorithm 3.1. Guidelines for practical choice of parameters is provided in Remark 4.3 in section 4.2. As can be seen, the algorithm is superficially complex, but, as in the case of line-search and trust-region methods that augment simple gradient-based methods in deterministic optimization, the relative lack of dependence on hyperparameters makes the algorithm robust and relatively easy to deploy.

Note that in Algorithm 3.1, and throughout the paper, we use $\tilde{x}_j$ to denote the iterate in the jth outer loop and $x_k^{(j)}$ to denote the iterate in the kth step of the jth inner loop.

Algorithm 3.1. SCSG for unconstrained finite-sum optimization.

Inputs: Number of stages T, initial iterate $\tilde{x}_0$, stepsizes $(\eta_j)_{j=1}^T$, block sizes $(B_j)_{j=1}^T$, inner loop sizes $(m_j)_{j=1}^T$, mini-batch sizes $(b_j)_{j=1}^T$.

Procedure

for $j = 1, 2, \ldots, T$ **do**
  Uniformly sample a batch $\mathcal{I}_j \subset \{1, \ldots, n\}$ with $|\mathcal{I}_j| = B_j$;
  $\mu_j \leftarrow \nabla f_{\mathcal{I}_j}(\tilde{x}_{j-1})$;
  $x_0^{(j)} \leftarrow \tilde{x}_{j-1}$;
  Generate $N_j \sim \mathrm{Geom}\left(\frac{m_j}{m_j + b_j}\right)$;
  for $k = 1, 2, \ldots, N_j$ **do**
    Uniformly sample a batch $\tilde{\mathcal{I}}_{k-1}^{(j)} \subset \{1, \ldots, n\}$ with $|\tilde{\mathcal{I}}_{k-1}^{(j)}| = b_j$;
    $\nu_{k-1}^{(j)} \leftarrow \nabla f_{\tilde{\mathcal{I}}_{k-1}^{(j)}}(x_{k-1}^{(j)}) - \nabla f_{\tilde{\mathcal{I}}_{k-1}^{(j)}}(x_0^{(j)}) + \mu_j$;
    $x_k^{(j)} \leftarrow x_{k-1}^{(j)} - \eta_j \nu_{k-1}^{(j)}$;
  end for
  $\tilde{x}_j \leftarrow x_{N_j}^{(j)}$;
end for

Output: $\tilde{x}_T$.

To measure the computational complexity of SCSG, let $T(\epsilon)$ denote the first time step at which $\tilde{x}_T$ is an ϵ-approximate solution, as well as all following iterates $\tilde{x}_{T+1}, \tilde{x}_{T+2}, \ldots$:

$$T(\epsilon) = \min\{T' : \mathbb{E}(f(\tilde{x}_T) - f(x^*)) \le \epsilon \ \forall T \ge T'\}. \tag{3.1}$$

This criterion is more stringent than those considered in other works that neglect the performance of $\tilde{x}_T$ for $T > T(\epsilon)$. The computational cost incurred for computing $\tilde{x}_T$ is

$$C_{\mathrm{comp}}(\epsilon) = \sum_{j=1}^{T(\epsilon)} (b_j N_j + B_j). \tag{3.2}$$

Noting that $C_{\text{comp}}(\epsilon)$ is random, we consider the average complexity obtained by taking the expectation of $C_{\text{comp}}(\epsilon)$. Since $N_j \sim \text{Geom}(\frac{m_j}{m_j+b_j})$, we have

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \sum_{j=1}^{T(\epsilon)} \left(b_j \frac{m_j}{b_j} + B_j \right) = \sum_{j=1}^{T(\epsilon)} (m_j + B_j). \tag{3.3}$$

3.1. Two key ideas: Adaptive batching and geometrization. The adaptivity of SCSG is achieved via two techniques: adaptive batching and geometrization. We provide intuitive motivation for these two ideas in this section.

The motivation for adaptive batching is straightforward. Heuristically, at the early stages of the optimization process, the iterate is far from the optimum, and a small subset of data is sufficient to reduce the variance. On the other hand, at later stages, finer variance reduction is required to prevent the iterate from moving in the wrong direction. By allowing the batch sizes to increase, SCSG behaves like SGD for the purpose of low-accuracy computation, while it behaves like SVRG for high-accuracy computation.

The motivation for geometrization is more subtle. To isolate its effect, let us consider a special case of SCSG in which the parameters are set as follows:

$$B_j = m_j \equiv n,\ b_j = 1,\ \eta_j \equiv \eta = O\left(\frac{1}{L}\right).$$

Note that the above setting is only used to illustrate the effect of geometrization, and the setting that leads to adaptivity to both strong convexity and target accuracy is more involved and given in section 4. In this simplified setting, SCSG reduces to SVRG if we replace line 5 by $N_j \sim \text{Unif}(\{0, \dots, m_j - 1\})$, with $m_j \equiv m$ for some positive integer m. (Although SVRG is usually implemented in practice by setting N_j to be a fixed m, a uniform random N_j is crucial for the analysis of SVRG [19].) SVRG achieves a rate of $O((n+\kappa)\log(1/\epsilon))$ only if

$$\frac{1}{\mu\eta(1-2\eta L)m} + \frac{2\eta L}{1-2\eta L} < 1. \tag{3.4}$$

This requires $m > \frac{1}{\mu\eta}$; hence, SVRG requires knowledge of μ to achieve the theoretical rate. We briefly sketch the step in the proof of the convergence of SVRG where this limitation arises, and we show how geometrization circumvents the need to know μ. To simplify our arguments we follow [19] and make the assumption that strong convexity holds everywhere for f; note that this is stronger than our assumption A2.

In Theorem 1 of [19], the following argument appears:

$$\begin{aligned} &2\eta\mathbb{E}\left\langle \nabla f(x_k^{(j)}), x_k^{(j)} - x^* \right\rangle - 4\eta^2 L\mathbb{E}(f(x_k^{(j)}) - f(x^*)) \\ &\le 4\eta^2 L\mathbb{E}(f(x_0^{(j)}) - f(x^*)) + \mathbb{E}\|x_k^{(j)} - x^*\|_2^2 - \mathbb{E}\|x_{k+1}^{(j)} - x^*\|_2^2. \end{aligned} \tag{3.5}$$

Strong convexity implies that

$$\begin{aligned} &2\eta(1-2\eta L)\mathbb{E}(f(x_k^{(j)}) - f(x^*)) + \eta\mu\mathbb{E}\|x_k^{(j)} - x^*\|_2^2 \\ &\le 4\eta^2 L\mathbb{E}(f(x_0^{(j)}) - f(x^*)) + \mathbb{E}\|x_k^{(j)} - x^*\|_2^2 - \mathbb{E}\|x_{k+1}^{(j)} - x^*\|_2^2. \end{aligned}$$

Note that this conclusion is independent of the choice of N_j and hence holds for both SVRG and SCSG. To assess the overall effect of the jth inner loop on the left-hand

side, we let $k = N_j$, thereby focusing on the last step of the inner loop, and we substitute $\tilde{x}_j$ for $x_{N_j}^{(j)}$ and $\tilde{x}_{j-1}$ for $x_0^{(j)}$. We have

$$\begin{aligned} &2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\eta\mu\mathbb{E}\|\tilde{x}_j-x^*\|_2^2 \\ &\le 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\mathbb{E}\|x_{N_j}^{(j)}-x^*\|_2^2-\mathbb{E}\|x_{N_j+1}^{(j)}-x^*\|_2^2. \end{aligned} \tag{3.6}$$

For SVRG, $N_j \sim \mathrm{Unif}\{0,\dots,m-1\}$, and thus (3.6) reduces to

$$\begin{aligned} &2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\eta\mu\mathbb{E}\|\tilde{x}_j-x^*\|_2^2 \\ &\le 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{1}{m}\mathbb{E}\|x_0^{(j)}-x^*\|_2^2-\frac{1}{m}\mathbb{E}\|x_m^{(j)}-x^*\|_2^2 \\ &= 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{1}{m}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2-\frac{1}{m}\mathbb{E}\|x_m^{(j)}-x^*\|_2^2. \end{aligned}$$

Unfortunately, given that $x_m^{(j)} \neq \tilde{x}_j$, the last two terms do not telescope, and one has to drop the final term, leading to the following conservative bound:

$$\begin{aligned} &2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\eta\mu\mathbb{E}\|\tilde{x}_j-x^*\|_2^2 \\ &\le 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{1}{m}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2. \end{aligned} \tag{3.7}$$

Without strong convexity (i.e., when $\mu = 0$), $\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2$ can be arbitrarily larger than $\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))$, and hence (3.7) is not helpful. Thus [19] exploits strong convexity at this point, using $\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2 \le \frac{2}{\mu}\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))$. Then (3.7) implies that

$$2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j)-f(x^*)) \le \left(4\eta^2 L+\frac{2}{m\mu}\right)\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*)).$$

This requires the coefficient on the left-hand side to be larger than that on the right-hand side, leading to the condition (3.4).

In summary, [19] relies on the knowledge of μ because it permits the removal of the last term in (3.6). By contrast, if N_j is a geometric random variable instead of a uniform random variable, the problem is completely circumvented by making use of the following elementary lemma.

LEMMA 3.1. *Let $N \sim \mathrm{Geom}(\gamma)$ for $\gamma > 0$. Then for any sequence $D_0, D_1, \dots$ with $\mathbb{E}|D_N| < \infty$,*

$$\mathbb{E}(D_N - D_{N+1}) = \left(\frac{1}{\gamma}-1\right)(D_0 - \mathbb{E}D_N).$$

Remark 3.2. The requirement $\mathbb{E}|D_N| < \infty$ is essential. A useful sufficient condition is $|D_k| = O(\mathrm{poly}(k))$ because a geometric random variable has finite moments of any order.

Proof. By definition,

$$\begin{aligned}\mathbb{E}(D_N - D_{N+1}) &= \sum_{n\geq 0}(D_n - D_{n+1})\gamma^n(1-\gamma)\\ &= (1-\gamma)\left(D_0 - \sum_{n\geq 1} D_n(\gamma^{n-1}-\gamma^n)\right) = (1-\gamma)\left(\frac{1}{\gamma}D_0 - \sum_{n\geq 0} D_n(\gamma^{n-1}-\gamma^n)\right)\\ &= (1-\gamma)\left(\frac{1}{\gamma}D_0 - \frac{1}{\gamma}\sum_{n\geq 0} D_n\gamma^n(1-\gamma)\right) = \left(\frac{1}{\gamma}-1\right)(D_0 - \mathbb{E}D_N),\end{aligned}$$

where the last equality is followed by the condition that $\mathbb{E}|D_N| < \infty$. ∎

Returning to (3.6) for SCSG with Lemma 3.1 in hand, where $N_j \sim \text{Geom}(\frac{n}{n+1})$ and $D_k = \mathbb{E}\|x_k^{(j)} - x^*\|_2^2$, and assuming that $\mathbb{E}D_{N_j} < \infty$, we obtain

$$\begin{aligned}&2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j) - f(x^*)) + \eta\mu\mathbb{E}\|\tilde{x}_j - x^*\|_2^2\\ &\leq 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1}) - f(x^*)) + \frac{1}{n}\mathbb{E}\|x_0^{(j)} - x^*\|_2^2 - \frac{1}{n}\mathbb{E}\|x_{N_j}^{(j)} - x^*\|_2^2\\ &= 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1}) - f(x^*)) + \frac{1}{n}\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 - \frac{1}{n}\mathbb{E}\|\tilde{x}_j - x^*\|_2^2. \qquad (3.8)\end{aligned}$$

The assumption that $\mathbb{E}D_{N_j} < \infty$ will be justified in our general theory and is taken for granted here to avoid distraction. Equation (3.8) can be rearranged to yield a function that provides a better assessment of progress than the function in (3.7):

$$\begin{aligned}&2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j) - f(x^*)) + \left(\frac{1}{n} + \eta\mu\right)\mathbb{E}\|\tilde{x}_j - x^*\|_2^2\\ &\leq 4\eta^2 L\mathbb{E}(f(\tilde{x}_{j-1}) - f(x^*)) + \frac{1}{n}\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2. \qquad (3.9)\end{aligned}$$

We accordingly view the left-hand side of (3.9) as a Lyapunov function and define

$$\mathcal{L}_j = 2\eta(1-2\eta L)\mathbb{E}(f(\tilde{x}_j) - f(x^*)) + \left(\frac{1}{n} + \eta\mu\right)\mathbb{E}\|\tilde{x}_j - x^*\|_2^2.$$

We then have

$$\mathcal{L}_j \leq \max\left\{\frac{2\eta L}{1-2\eta L}, \frac{1}{1+n\eta\mu}\right\}\mathcal{L}_{j-1} \triangleq \lambda^{-1}\mathcal{L}_{j-1}.$$

As a result,

$$\mathcal{L}_T \leq \epsilon \quad \forall T \geq \frac{\log\frac{\mathcal{L}_0}{\epsilon}}{\log\lambda} \Longrightarrow T(\epsilon) \leq \frac{\log\frac{\mathcal{L}_0}{\epsilon}}{\log\lambda},$$

and, by (3.2),

$$\mathbb{E}C_{\text{comp}}(\epsilon) \leq 2nT(\epsilon) = O\left(n\frac{\log\frac{\mathcal{L}_0}{\epsilon}}{\log\lambda}\right).$$

Suppose $\eta L < \frac{1}{6}$, then

$$\lambda \geq 1 + (n\eta\mu \wedge 1) \Longrightarrow \frac{1}{\log\lambda} = O\left(\frac{1}{n\eta\mu\wedge 1}\right) = O\left(\frac{\kappa}{n} + 1\right).$$

Therefore the complexity of SCSG is

$$\mathbb{E}C_{\text{comp}}(\epsilon) = O\left((n+\kappa)\log\left(\frac{\mathcal{L}_0}{\epsilon}\right)\right).$$

In summary, the better control provided by geometrization enables SCSG to achieve the fast rate of SVRG without knowledge of μ.

4. Convergence analysis of SCSG for unregularized smooth problems.

4.1. One-epoch analysis. We start with the analysis for a single epoch. The key difficulty lies in controlling the bias of $\nu_k^{(j)}$, conditional on $\mathcal{I}_j$ drawn at the beginning of the jth epoch. We have

$$\mathbb{E}_{\tilde{\mathcal{I}}_k^{(j)}}\nu_k^{(j)} - \nabla f(x_k^{(j)}) = \nabla f_{\mathcal{I}_j}(\tilde{x}_{j-1}) - \nabla f(\tilde{x}_{j-1}) \triangleq e_j. \tag{4.1}$$

We deal with this extra bias by exploiting Lemma 3.1 and obtaining the following theorem which connects the iterates produced in consecutive epochs. The proof of the theorem is relegated to section 4.5.

THEOREM 4.1. *Fix any* $\Gamma \le 1/4$. *Assume that*

$$\eta_j L \le \min\left\{\frac{1-\Gamma}{2}, \Gamma b_j, \frac{\Gamma^2 b_j B_j}{2m_j}\right\}, \quad m_j \ge b_j. \tag{4.2}$$

Then under assumptions A1 *and* A2,

$$\begin{aligned}&\mathbb{E}(f(\tilde{x}_j) - f(x^*)) + \left(\frac{2b_j(1-\Gamma)}{3\eta_j m_j} + \frac{\mu}{6}\right)\mathbb{E}\|\tilde{x}_j - x^*\|_2^2 \\ &\le 4\Gamma\mathbb{E}(f(\tilde{x}_{j-1}) - f(x^*)) + \frac{2b_j(1-\Gamma)}{3\eta_j m_j}\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 + \frac{2\eta_j L}{\Gamma b_j}\frac{m_j}{B_j}D_H I(B_j < n).\end{aligned}$$

4.2. Multi-epoch analysis. We now turn to the multi-epoch analysis, focusing on using the one-epoch analysis to determine the setting of the hyperparameters. Interestingly, we require that the batch size B_j scales as the square of the number of inner-loop iterations m_j. The proof is relegated to subsection 4.5.2.

THEOREM 4.2. *Fix any constant* $\alpha > 1$, $m_0 > 0$, *and* $\xi \in (0,1)$. *Let*

$$\eta_j \equiv \eta, \quad b_j \equiv b, \quad m_j = m_0\alpha^j, \quad B_j = \lceil B_0\alpha^{2j} \wedge n\rceil.$$

Take $\Gamma = 1/4\alpha^{1/\xi}$ *and assume that*

$$m_0 \ge b/\Gamma \quad 2\eta L \le \min\left\{1-\Gamma, 2\Gamma b, \Gamma^2 b B_0/m_0\right\}.$$

Then

$$\mathbb{E}(f(\tilde{x}_T) - f(x^*)) \le \Lambda_T^{-1}\frac{D_x}{2\eta L} + \tilde{\Lambda}_T^{-1}\frac{2\eta L m_0}{\Gamma b B_0}D_H(T \wedge T_n^*),$$

where $\Lambda_T = \lambda_T\lambda_{T-1}\cdots\lambda_1$, $\tilde{\Lambda}_T = \tilde{\lambda}_T\tilde{\lambda}_{T-1}\cdots\tilde{\lambda}_1$,

$$\lambda_j = \begin{cases}\alpha & (j \le T_\kappa^*),\\ \alpha^{1/\xi} & (j > T_\kappa^*),\end{cases} \qquad \tilde{\lambda}_j = \begin{cases}\alpha & (j \le T_n^* \vee T_\kappa^*),\\ \alpha^{1/\xi} & (j > T_n^* \vee T_\kappa^*),\end{cases}$$

and let T_κ^*, T_n^* *be positive numbers such that*

$$\alpha^{T_\kappa^*} = \frac{1}{\eta\mu}, \quad B_0\alpha^{2T_n^*} = n.$$

Remark 4.3. In practice, we recommend the following setting as a default:

$$\alpha = 1.25, \quad m_0 = 5B_0 = 50b.$$

This setting works well as demonstrated in Appendix D in [26]. For those examples, b was chosen as $10^{-4}n$ for fair comparison. Here we choose B_0 smaller than m_0 because the batch size B_t grows faster than the inner-loop size m_t, and the variance reduction is more effective in later stages. Under this setting, SCSG takes $\lceil \log(n/B_0)/2\log\alpha \rceil = 16$ passes for B_t to reach n. This creates a reasonably long transition from little to full variance reduction.

4.3. Complexity analysis. Under the specification of Theorem 4.2 and recalling the definition of $T(\epsilon)$ in (3.1), we have

$$\sum_{j=1}^{T(\epsilon)} m_j = m_0 \sum_{j=1}^{T(\epsilon)} \alpha^j = O\left(\alpha^{T(\epsilon)}\right).$$

On the other hand,

$$\sum_{j=1}^{T(\epsilon)} B_j = O\left(\sum_{j=1}^{T(\epsilon)} (\alpha^{2j} \wedge n)\right) = O\left(\min\left\{\sum_{j=1}^{T(\epsilon)} \alpha^{2j}, \sum_{j=1}^{T(\epsilon)} n\right\}\right) = O\left(\alpha^{2T(\epsilon)} \wedge nT(\epsilon)\right).$$

By (3.3), we conclude that

$$\mathbb{E}C_{\mathrm{comp}}(\epsilon) = O\left(\alpha^{T(\epsilon)} + \alpha^{2T(\epsilon)} \wedge nT(\epsilon)\right) = O\left(\alpha^{2T(\epsilon)} \wedge (\alpha^{T(\epsilon)} + nT(\epsilon))\right). \tag{4.3}$$

The following theorem gives the size of $T(\epsilon)$ and thus provides the theoretical complexity of SCSG. The proof is relegated to subsection 4.5.3.

THEOREM 4.4. *Under the specification of Theorem* 4.2*, we have*

$$\mathbb{E}C_{\mathrm{comp}}(\epsilon) = O\left(A(\epsilon)^2 \wedge (A(\epsilon) + n\log A(\epsilon))\right),$$

where

$$A(\epsilon) = \tilde{O}\left(\min\left\{\frac{D}{\epsilon}, \kappa\left(\frac{D_x}{\epsilon\kappa}\right)_*^{\xi} + \frac{D_H}{\epsilon}, \tilde{\kappa}\left(\frac{D}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}\right), \quad \tilde{\kappa} = \sqrt{n} + \kappa.$$

In particular,

$$\mathbb{E}C_{\mathrm{comp}}(\epsilon) = \tilde{O}\left(\min\left\{\frac{D^2}{\epsilon^2}, \frac{D_H^2}{\epsilon^2} + \kappa^2\left(\frac{D_x}{\epsilon\kappa}\right)_*^{2\xi}, n + \frac{D}{\epsilon}, n + \tilde{\kappa}\left(\frac{D}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}\right). \tag{4.4}$$

Remark 4.5. The version of SCSG considered in Theorem 4.4 that achieves the complexity (4.4) is ϵ-independent and almost universal.

4.4. Discussion. Our complexity result involves the unusual terms $\left(\frac{D_x}{\epsilon\kappa}\right)_*^{2\xi}$ and $\left(\frac{D}{\epsilon\kappa}\right)_*^{2\xi}$. However, they are relatively insignificant as the exponent ξ can be made arbitrarily small, and $\frac{1}{\epsilon\kappa}$ is small in practice unless the target accuracy is unusually high. For instance, under the setting given in Remark 4.3, ξ can be as small as $\log\alpha/\log(m_0/4b) \approx 0.088$ to guarantee the condition $m_0 \geq b/\Gamma$ as long as the stepsize

is sufficiently small. Thus, the term $\left(\frac{D_x}{\epsilon\kappa}\right)_*^{2\xi}$ is generally negligible. If we use $\tilde{\tilde{O}}$ to denote a bound that hides these terms and the logarithmic terms, we have

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\frac{D^2}{\epsilon^2} \wedge \left\{\kappa^2 + \frac{D_H^2}{\epsilon^2}\right\} \wedge \left\{n + \frac{D}{\epsilon}\right\} \wedge \{n+\kappa\}\right). \tag{4.5}$$

We discuss some of the consequences of (4.5).

4.4.1. Adaptivity to target accuracy. For non–strongly convex objectives, (4.5) implies that

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\frac{D^2}{\epsilon^2} \wedge \left\{n + \frac{D}{\epsilon}\right\}\right), \tag{4.6}$$

whereas, for strongly convex objectives, (4.5) implies that

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\left\{\kappa^2 + \frac{D_H^2}{\epsilon^2}\right\} \wedge \{n+\kappa\}\right). \tag{4.7}$$

Both (4.6) and (4.7) exhibit the adaptivity of SCSG to the target accuracy: for low-accuracy computation (i.e., large ϵ), SCSG achieves the same complexity as SGD for non–strongly convex objectives, which can be much more efficient than SVRG-type algorithms in the setting of large datasets (i.e., large n). On the other hand, for high-accuracy computation (i.e., small ϵ), SCSG avoids the high variance of SGD and achieves the same complexity as SVRG++ [5] for non–strongly convex objectives and as SVRG for strongly convex objectives [19].

4.4.2. Adaptivity to strong convexity. The first two terms of (4.5) are independent of n:

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\frac{D^2}{\epsilon^2} \wedge \left\{\kappa^2 + \frac{D_H^2}{\epsilon^2}\right\}\right). \tag{4.8}$$

The last two terms of (4.5) depend on n but have better dependence on ϵ:

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\left\{n + \frac{D}{\epsilon}\right\} \wedge \{n+\kappa\}\right). \tag{4.9}$$

Both (4.8) and (4.9) show the adaptivity of SCSG to strong convexity. In both cases, if $\kappa \ll \frac{1}{\epsilon}$, SCSG benefits from the strong convexity: for the former, (4.8) yields

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\kappa^2 + \frac{D_H^2}{\epsilon^2}\right),$$

which can be much smaller than $O\left(\frac{D^2}{\epsilon^2}\right)$ if $D_x \gg D_H$. For the latter, (4.9) yields

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(n+\kappa\right),$$

which is the same as SVRG but without the knowledge of μ. On the other hand, in ill-conditioned problems where $\kappa \gg \frac{1}{\epsilon}$, SCSG still achieves the best of SGD and SVRG++ [5] for non–strongly convex objectives. This is not achieved by adaptive SVRG [53] and is only partially achieved by randomized SVRG [25], which requires two sequences of iterates. Finally, although SAG and SAGA provide guarantees in ill-conditioned problems, they have an inferior complexity of $\tilde{O}\left(\frac{n}{\epsilon} \wedge (n+\kappa)\right)$.

4.4.3. Weaker requirement on gradients. For algorithms without access to full gradients, it is necessary to impose some conditions on $\nabla f_i(x)$. The strongest condition imposes a uniform bound (see, e.g., [34]):

$$\sigma^2 \triangleq \max_i \sup_x \|\nabla f_i(x)\|_2^2 < \infty, \tag{4.10}$$

while a slightly milder condition imposes the following bound (see, e.g., [29]):

$$A^2 \triangleq \sup_x \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x) - \nabla f(x)\|_2^2 < \infty. \tag{4.11}$$

These two types of conditions are typical in analyses of SGD when f_i is not assumed to be convex. This is satisfied by many practical problems, e.g., generalized linear models. In our situation, the extra assumption on the convexity of each component allows us to relax assumptions such as (4.10) and (4.11) into

$$\mathcal{H} \triangleq \frac{1}{n}\sum_{i=1}^n \|\nabla f_i(x^*)\|_2^2 < \infty.$$

First, it is easy to show that

$$\mathcal{H} \le A^2 \le \sigma^2.$$

More importantly, $\mathcal{H}$ can be much smaller than the other two measures, and there are common situations where $A^2 = \sigma^2 = \infty$ while $\mathcal{H} < \infty$. For instance, in least-squares problems where $f_i(x) = \frac{1}{2}(a_i^T x - b_i)^2$, $A^2 = \sigma^2 = \infty$ unless the domain is bounded. Although assuming a bounded domain is common in the literature, there is generally no guarantee, at least for algorithms involving stochasticity, that the iterate will stay in the domain unless a projection step is performed. However, the projection step is never performed in practice, and thus the bounded domain assumption is artificial. By contrast, [25] show that

$$\mathcal{H} \le \frac{2}{n}\sum_{i=1}^n (2b_i^2 - f_i(x^*)) \le \frac{4\sum_{i=1}^n b_i^2}{n}$$

for least-squares problems, without a bounded domain. This implies that $\mathcal{H} = O(1)$ provided that $\frac{1}{n}\sum_{i=1}^n b_i^2 = O(1)$. Similar bounds can be derived for generalized linear models [25]. It turns out that $\mathcal{H} = O(1)$ for various applications where there is no guarantee for σ^2 or A^2. We refer the readers to [25] for an extensive discussion.

4.4.4. Optimality of the complexity bound. To the best of our knowledge, SCSG is the first stochastic algorithm that achieves adaptivity to both target accuracy and strong convexity. However, it is still illuminating to compare each component of (4.5) separately with the best achievable rate in the literature.

- The first component $\tilde{\tilde{O}}\left(\frac{D^2}{\epsilon^2}\right)$ is optimal in terms of ϵ-dependence for non–strongly convex objectives [2, 52]. Under slightly stronger assumptions on the gradient bounds (but without the convexity of each f_i), mini-batched SGD achieves the $O\left(\frac{1}{\epsilon^2}\right)$ rate [34, 29]. However, the dependence on D is suboptimal. Without knowing D_x and σ^2 or A^2, defined in (4.10) and (4.11), the resulting complexity of (mini-batched) SGD can be no better than

$O\left(\frac{(D_x \vee A)^2}{\epsilon^2}\right) \geq O\left(\frac{D^2}{\epsilon^2}\right)$. When they are known, [34, 29] are able to improve it to $O\left(\frac{D_x \sigma}{\epsilon^2}\right)$ and $O\left(\frac{D_x A}{\epsilon^2}\right)$. With averaging, [20] improves it to

$$O\left(\frac{D_x}{\epsilon} + \frac{\sigma^2}{\epsilon^2}\right) \geq O\left(\frac{D_x}{\epsilon} + \frac{D_H^2}{\epsilon^2}\right).$$

With momentum acceleration, [22] further improves the rate to

$$O\left(\sqrt{\frac{D_x}{\epsilon}} + \frac{\sigma^2}{\epsilon^2}\right) \geq O\left(\sqrt{\frac{D_x}{\epsilon}} + \frac{D_H^2}{\epsilon^2}\right). \tag{4.12}$$

- To the best of our knowledge, the second component $\tilde{\tilde{O}}\left(\kappa^2 + \frac{D_H^2}{\epsilon^2}\right)$ is new. When μ is known and $\mathbb{E}\|\nabla f_i(x)\|_2^2$ is uniformly bounded for all i and x, a requirement that is more stringent than our setting, it is known that the optimal complexity in terms of ϵ-dependence and μ-dependence is $O\left(\frac{\kappa}{\epsilon}\right)$; see, e.g., [18, 48] for the upper bound and [52] for the lower bound. However, the lower bound is established under the condition that μ is known. It remains an interesting direction to derive a tight lower bound when μ is unknown.
- The third component $\tilde{\tilde{O}}\left(n + \frac{D}{\epsilon}\right)$ should be suboptimal in terms of both ϵ and D. SVRG++ [5] achieves the $\tilde{O}\left(n + \frac{D_x}{\epsilon}\right)$ rate. By adding momentum terms, Adaptive SVRG [38] slightly improves the rate to

$$\tilde{O}\left(\left\{n + \frac{D_x}{\epsilon}\right\} \wedge n\sqrt{\frac{D_x}{\epsilon}}\right).$$

  On the other hand, [52] prove a lower bound

$$\tilde{O}\left(n + \sqrt{\frac{nD_x}{\epsilon}}\right).$$

  This can be achieved by accelerated SDCA [47] or Katyusha [3]. However, the former has only been established for particular problems, such as generalized linear models, and the latter involves black-box acceleration [4], which requires setting the parameters based on unknown quantities, such as D_x. The Varag algorithm [24], which appeared after our work, is the first algorithm to achieve the lower bound for generic finite-sum optimization problems.
- The last component $\tilde{\tilde{O}}\left(n + \kappa\right)$ has been proved by [6] to be optimal, up to small factors $\left(\frac{D_H}{\epsilon\kappa}\right)_*^{2\xi}$, for a large class of algorithms when μ is unknown. The story is different when μ is known. The optimal complexity can be improved to $\tilde{O}(n + \sqrt{n\kappa})$ and can be achieved, for instance, by Katyusha [3].

In summary, a major remaining challenge is to derive oracle lower bounds involving all of ϵ, n, D_x, D_H for ϵ-independent and almost universal algorithms.

4.5. Proofs.

4.5.1. Lemmas. We start by presenting four lemmas, with proofs given in Appendix A of [26]. The first lemma gives an upper bound of the expected squared norm of $\nu_k^{(j)}$, which is standard in the analyses of most first-order methods.

LEMMA 4.6. *Under assumption* A1*,*

$$\begin{aligned}&\mathbb{E}_{\tilde{\mathcal{I}}_k^{(j)}}\|\nu_k^{(j)}\|_2^2\\&\le \frac{2L}{b_j}(f(x_0^{(j)})-f(x_k^{(j)}))+\frac{2L}{b_j}\left\langle\nabla f(x_k^{(j)}),x_k^{(j)}-x_0^{(j)}\right\rangle+2\|\nabla f(x_k^{(j)})\|_2^2+2\|e_j\|_2^2,\end{aligned}$$

where e_j *is defined as in* (4.1)*.*

The second lemma gives an upper bound for $\mathbb{E}\|e_j\|_2^2$.

LEMMA 4.7. *Under assumption* A1*,*

$$\mathbb{E}\|e_j\|_2^2 \le \frac{2}{B_j}\left\{2L\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\mathcal{H}\,I(B_j<n)\right\}.$$

The third lemma below connects the iterates $\tilde{x}_j$ and $\tilde{x}_{j-1}$ in adjacent epochs. The proof exploits the elegant property of geometrization.

LEMMA 4.8. *Let* $u\in\mathbb{R}^d$ *be any variable that is independent of* $\mathcal{I}_j$ *and subsequent random subsets within the* j*th epoch,* $\tilde{\mathcal{I}}_0^{(j)},\tilde{\mathcal{I}}_1^{(j)},\ldots$*, with* $\mathbb{E}\|u-x^*\|_2^2<\infty$*. Then under assumption* A1*,*

$$\begin{aligned}&2\eta_j\mathbb{E}\left\langle\nabla f(\tilde{x}_j),\tilde{x}_j-u\right\rangle\\&\le\frac{b_j}{m_j}\left(\mathbb{E}\|\tilde{x}_{j-1}-u\|_2^2-\mathbb{E}\|\tilde{x}_j-u\|_2^2\right)+2\eta_j\mathbb{E}\left\langle e_j,\tilde{x}_{j-1}-\tilde{x}_j\right\rangle+\mathbb{E}W_j,\end{aligned}$$

where

$$W_j=\frac{2\eta_j^2L}{b_j}(f(\tilde{x}_{j-1})-f(\tilde{x}_j))+\frac{2\eta_j^2L}{b_j}\left\langle\nabla f(\tilde{x}_j),\tilde{x}_j-\tilde{x}_{j-1}\right\rangle+2\eta_j^2\|\nabla f(\tilde{x}_j)\|_2^2+2\eta_j^2\|e_j\|_2^2.$$

The term $\mathbb{E}\left\langle e_j,\tilde{x}_{j-1}-\tilde{x}_j\right\rangle$ is nonstandard. We derive an upper bound in the following lemma. Surprisingly, this lemma is a direct consequence of Lemma 4.8.

LEMMA 4.9. *Fix any* $\gamma_j>0$*. Under assumption* A1*,*

$$\begin{aligned}&2\eta_j\mathbb{E}\left\langle e_j,\tilde{x}_{j-1}-\tilde{x}_j\right\rangle\\&\le-2\gamma_j\eta_j\mathbb{E}\left\langle\nabla f(\tilde{x}_j),\tilde{x}_j-\tilde{x}_{j-1}\right\rangle+\gamma_j\mathbb{E}W_j+\frac{\eta_j^2m_j}{b_j}\frac{(1+\gamma_j)^2}{\gamma_j}\mathbb{E}\|e_j\|_2^2.\end{aligned}$$

4.5.2. Proof of Theorem 4.1. We start from a more general version of Theorem 4.1.

THEOREM 4.10. *Fix any* $\Gamma_j\in(0,1)$*. Assume that*

$$\eta_jL\le\min\left\{\frac{1-\Gamma_j}{2},\Gamma_jb_j\right\}. \tag{4.13}$$

Then under assumption A1*,*

$$\begin{aligned}&\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\frac{b_j(1-\Gamma_j)}{2\eta_jm_j}\mathbb{E}\|\tilde{x}_j-x^*\|_2^2\\&\le\left(\Gamma_j+2\left(\frac{1}{\Gamma_j}+\frac{2b_j}{m_j}\right)\frac{\eta_jLm_j}{b_jB_j}\right)\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{b_j(1-\Gamma_j)}{2\eta_jm_j}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2\\&\quad+\left(\frac{1}{\Gamma_j}+\frac{2b_j}{m_j}\right)\frac{\eta_jm_j}{b_jB_j}\mathcal{H}I(B_j<n).\end{aligned}$$

Proof. Letting $u = x^*$ in Lemma 4.8 and applying Lemma 4.9 with $\gamma_j = \Gamma_j/(1-\Gamma_j)$ (i.e. $\Gamma_j = \gamma_j/(1+\gamma_j)$), we obtain

$$\begin{aligned}
2\eta_j \mathbb{E}\left\langle \nabla f(\tilde{x}_j), \tilde{x}_j - x^* \right\rangle &\le \frac{b_j}{m_j}\left(\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 - \mathbb{E}\|\tilde{x}_j - x^*\|_2^2\right) \\
&\quad - 2\gamma_j\eta_j \mathbb{E}\left\langle \nabla f(\tilde{x}_j), \tilde{x}_j - \tilde{x}_{j-1}\right\rangle + (1+\gamma_j)\mathbb{E}W_j + \frac{\eta_j^2 m_j}{b_j}\frac{(1+\gamma_j)^2}{\gamma_j}\mathbb{E}\|e_j\|_2^2 \\
&\le \frac{b_j}{m_j}\left(\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 - \mathbb{E}\|\tilde{x}_j - x^*\|_2^2\right) + 2\eta_j^2(1+\gamma_j)\mathbb{E}\|\nabla f(\tilde{x}_j)\|_2^2 \\
&\quad + \frac{2(1+\gamma_j)\eta_j^2 L}{b_j}\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)) \\
&\quad - 2\eta_j\left(\gamma_j - \frac{(1+\gamma_j)\eta_j L}{b_j}\right)\mathbb{E}\left\langle \nabla f(\tilde{x}_j), \tilde{x}_j - \tilde{x}_{j-1}\right\rangle \\
&\quad + \frac{(1+\gamma_j)\eta_j^2 m_j}{b_j}\left(\frac{1+\gamma_j}{\gamma_j} + \frac{2b_j}{m_j}\right)\mathbb{E}\|e_j\|_2^2. \qquad (4.14)
\end{aligned}$$

First, by Lemma C.1 in [26], with $x = x^*, y = \tilde{x}_j$ and the fact that $\nabla f(x^*) = 0$,

$$\|\nabla f(\tilde{x}_j)\|_2^2 \le 2L(f(x^*) - f(\tilde{x}_j) + \langle \nabla f(\tilde{x}_j), \tilde{x}_j - x^*\rangle). \qquad (4.15)$$

Second, since $\gamma_j \ge \frac{(1+\gamma_j)\eta_j L}{b_j}$ by (4.13), by convexity of f we obtain that

$$\begin{aligned}
&\frac{2(1+\gamma_j)\eta_j^2 L}{b_j}\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)) - 2\eta_j\left(\gamma_j - \frac{(1+\gamma_j)\eta_j L}{b_j}\right)\mathbb{E}\left\langle \nabla f(\tilde{x}_j), \tilde{x}_j - \tilde{x}_{j-1}\right\rangle \\
&\le \frac{2(1+\gamma_j)\eta_j^2 L}{b_j}\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)) - 2\eta_j\left(\gamma_j - \frac{(1+\gamma_j)\eta_j L}{b_j}\right)\mathbb{E}(f(\tilde{x}_j) - f(\tilde{x}_{j-1})) \\
&= 2\eta_j\gamma_j\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)). \qquad (4.16)
\end{aligned}$$

Combining (4.14)–(4.16) yields

$$\begin{aligned}
&2\eta_j(1 - 2(1+\gamma_j)\eta_j L)\mathbb{E}\left\langle \nabla f(\tilde{x}_j), \tilde{x}_j - x^*\right\rangle + 4(1+\gamma_j)\eta_j^2 L\mathbb{E}(f(\tilde{x}_j) - f(x^*)) \\
&\quad \le \frac{b_j}{m_j}\left(\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 - \mathbb{E}\|\tilde{x}_j - x^*\|_2^2\right) + 2\eta_j\gamma_j\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)) \\
&\quad\quad + \frac{(1+\gamma_j)\eta_j^2 m_j}{b_j}\left(\frac{1+\gamma_j}{\gamma_j} + \frac{2b_j}{m_j}\right)\mathbb{E}\|e_j\|_2^2. \qquad (4.17)
\end{aligned}$$

Since $1 \ge 2(1+\gamma_j)\eta_j L$ by (4.13), by convexity of f,

$$\langle \nabla f(\tilde{x}_j), \tilde{x}_j - x^*\rangle \ge f(\tilde{x}_j) - f(x^*).$$

By (4.17) and Lemma 4.7,

$$\begin{aligned}
&2\eta_j\mathbb{E}(f(\tilde{x}_j) - f(x^*)) \\
&\le \frac{b_j}{m_j}\left(\mathbb{E}\|\tilde{x}_{j-1} - x^*\|_2^2 - \mathbb{E}\|\tilde{x}_j - x^*\|_2^2\right) + 2\eta_j\gamma_j\mathbb{E}(f(\tilde{x}_{j-1}) - f(\tilde{x}_j)) \\
&\quad + 2(1+\gamma_j)\left(\frac{1+\gamma_j}{\gamma_j} + \frac{2b_j}{m_j}\right)\frac{\eta_j^2 m_j}{b_j B_j}\left(2L\mathbb{E}(f(\tilde{x}_{j-1}) - f(x^*)) + \mathcal{H}I(B_j < n)\right).
\end{aligned}$$

Rearranging the terms, we have

$$\begin{aligned}
&2\eta_j(1+\gamma_j)\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\frac{b_j}{m_j}\mathbb{E}\|\tilde{x}_j-x^*\|_2^2\\
&\le 2\eta_j\left(\gamma_j+2(1+\gamma_j)\left(\frac{1+\gamma_j}{\gamma_j}+\frac{2b_j}{m_j}\right)\frac{\eta_j L m_j}{b_j B_j}\right)\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))\\
&\quad+\frac{b_j}{m_j}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2+2(1+\gamma_j)\left(\frac{1+\gamma_j}{\gamma_j}+\frac{2b_j}{m_j}\right)\frac{\eta_j^2 m_j}{b_j B_j}\mathcal{H}I(B_j<n).
\end{aligned}$$

Dividing both sides by $2\eta_j(1+\gamma_j)$ and recalling the definition that $\Gamma_j=\gamma_j/(1+\gamma_j)$, we complete the proof. ∎

Proof of Theorem 4.1. Let $\Gamma_j\equiv\Gamma$ as in Theorem 4.10. Under condition (4.2), (4.13) is satisfied because $\Gamma\le 1/4$. Moreover,

$$\Gamma+\left(\frac{1}{\Gamma}+\frac{2b_j}{m_j}\right)\frac{2\eta_j L m_j}{b_j B_j}\le\Gamma+\left(\frac{1}{\Gamma}+2\right)\Gamma^2\le 3\Gamma$$

and

$$\frac{1}{\Gamma}+\frac{2b_j}{m_j}\le\frac{1}{\Gamma}+2\le\frac{3}{2\Gamma}.$$

By assumption A2,

$$\frac{1}{4}(f(\tilde{x}_j)-f(x^*))\ge\frac{\mu}{8}\|\tilde{x}_j-x^*\|_2^2.$$

By Theorem 4.10 we have

$$\begin{aligned}
&\frac{3}{4}\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\left(\frac{b_j(1-\Gamma)}{2\eta_j m_j}+\frac{\mu}{8}\right)\mathbb{E}\|\tilde{x}_j-x^*\|_2^2\\
&\le 3\Gamma\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{b_j(1-\Gamma)}{2\eta_j m_j}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2+\frac{3}{2\Gamma}\frac{\eta_j m_j}{b_j B_j}\mathcal{H}I(B_j<n).
\end{aligned}$$

The proof is completed by multiplying by 4/3 on both sides and by the fact that $D_H=\mathcal{H}/L$. ∎

4.5.3. Other proofs.

Proof of Theorem 4.2. By Theorem 4.1 with $\Gamma=1/4\alpha^{1/\xi}$,

$$\begin{aligned}
&\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\left(\frac{2b(1-\Gamma)}{3\eta m_j}+\frac{\mu}{6}\right)\mathbb{E}\|\tilde{x}_j-x^*\|_2^2\\
&\le\frac{1}{\alpha^{1/\xi}}\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{2b(1-\Gamma)}{3\eta m_j}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2+\frac{2\eta L}{\Gamma b}\frac{m_j}{B_j}D_H I(B_j<n)\\
&=\frac{1}{\alpha^{1/\xi}}\mathbb{E}(f(\tilde{x}_{j-1})-f(x^*))+\frac{2b(1-\Gamma)}{3\eta m_j}\mathbb{E}\|\tilde{x}_{j-1}-x^*\|_2^2 n+\frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*).
\end{aligned}$$

Let

$$\mathcal{L}_j=\mathbb{E}(f(\tilde{x}_j)-f(x^*))+\left(\frac{2b(1-\Gamma)}{3\eta m_j}+\frac{\mu}{6}\right)\mathbb{E}\|\tilde{x}_j-x^*\|_2^2.$$

Then

$$\begin{aligned}\mathcal{L}_j &\le \max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{\frac{2b(1-\Gamma)}{3\eta m_j}}{\frac{2b(1-\Gamma)}{3\eta m_{j-1}}+\frac{\mu}{6}}\right\}\mathcal{L}_{j-1} + \frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*)\\ &= \max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\frac{m_j}{m_{j-1}}+\frac{\eta\mu m_j}{4b(1-\Gamma)}}\right\}\mathcal{L}_{j-1} + \frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*)\\ &= \max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha+\eta\mu\alpha^j\frac{m_0}{4b(1-\Gamma)}}\right\}\mathcal{L}_{j-1} + \frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*)\\ &\le \max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha+\eta\mu\alpha^{j+1/\xi}}\right\}\mathcal{L}_{j-1} + \frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*), \qquad (4.18)\end{aligned}$$

where the last line uses the condition that

$$\frac{m_0}{4b(1-\Gamma)} \ge \frac{m_0}{4b} \ge \frac{1}{4\Gamma} = \alpha^{1/\xi}.$$

For any $j \ge 0$,

$$\max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha+\eta\mu\alpha^{j+1/\xi}}\right\} \le \max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha}\right\} \le \frac{1}{\alpha}.$$

When $j \ge T_\kappa^*$, we have $\alpha^j \ge \kappa/\eta L = 1/\eta\mu$, and thus

$$\max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha+\eta\mu\alpha^{j+1/\xi}}\right\} \le \frac{1}{\alpha^{1/\xi}}.$$

In summary,

$$\max\left\{\frac{1}{\alpha^{1/\xi}}, \frac{1}{\alpha+\eta\mu\alpha^{j+1/\xi}}\right\} \le \lambda_j^{-1}.$$

Plugging this into (4.18), we conclude that

$$\mathcal{L}_j \le \lambda_j^{-1}\mathcal{L}_{j-1} + \frac{2\eta L m_0}{\Gamma b B_0}\frac{1}{\alpha^j}D_H I(j<T_n^*). \qquad (4.19)$$

Finally, we prove the following statement by induction:

$$\mathcal{L}_T \le \Lambda_T^{-1}\mathcal{L}_0 + C_0\tilde{\Lambda}_T^{-1}D_H(T\wedge T_n^*), \quad \text{where } C_0 = \frac{2\eta L m_0}{\Gamma b B_0}. \qquad (4.20)$$

It is obvious that (4.20) holds for $T=0$. Suppose it holds for $T-1$; then by (4.19),

$$\begin{aligned}\mathcal{L}_T &\le \lambda_T^{-1}\mathcal{L}_{T-1} + C_0\frac{D_H}{\alpha^T}I(T<T_n^*)\\ &\le \lambda_T^{-1}\left(\Lambda_{T-1}^{-1}\mathcal{L}_0 + C_0\tilde{\Lambda}_{T-1}^{-1}D_H((T-1)\wedge T_n^*)\right) + C_0\frac{D_H}{\alpha^T}I(T<T_n^*)\\ &= \Lambda_T^{-1}\mathcal{L}_0 + C_0 D_H\left(\lambda_T^{-1}\tilde{\Lambda}_{T-1}^{-1}((T-1)\wedge T_n^*) + \alpha^{-T}I(T<T_n^*)\right)\\ &\le \Lambda_T^{-1}\mathcal{L}_0 + C_0 D_H\left(\tilde{\Lambda}_T^{-1}((T-1)\wedge T_n^*) + \alpha^{-T}I(T<T_n^*)\right),\end{aligned}$$

where the last line uses the fact that $\tilde{\lambda}_T \le \lambda_T$ for all $T > 0$. If $T < T_n^*$, then $\tilde{\Lambda}_T = \alpha^{-T}$, and thus

$$\tilde{\Lambda}_T^{-1}((T-1) \wedge T_n^*) + \alpha^{-T} I(T < T_n^*) = \tilde{\Lambda}_T^{-1}(T \wedge T_n^*).$$

If $T > T_n^*$,

$$\tilde{\Lambda}_T^{-1}((T-1) \wedge T_n^*) + \alpha^{-T} I(T < T_n^*) = \tilde{\Lambda}_T^{-1}((T-1) \wedge T_n^*) \le \tilde{\Lambda}_T^{-1}(T \wedge T_n^*).$$

Therefore, (4.20) is proved. The proof is then completed by noting that

$$\mathcal{L}_T \ge \mathbb{E}(f(\tilde{x}_T) - f(x^*))$$

and

$$\begin{aligned}\mathcal{L}_0 &= \mathbb{E}(f(\tilde{x}_0) - f(x^*)) + \left(\frac{2b(1-\Gamma)}{3\eta m_0} + \frac{\mu}{6}\right)\mathbb{E}\|\tilde{x}_0 - x^*\|_2^2 \\ &\overset{\text{(i)}}{\le} \left(\frac{1}{2} + \frac{2b(1-\Gamma)}{3\eta L m_0} + \frac{\mu}{6L}\right) D_x \overset{\text{(ii)}}{\le} \left(\frac{2}{3} + \frac{1}{6\eta L}\right) D_x \\ &\overset{\text{(iii)}}{\le} \left(\frac{1}{4\eta L} + \frac{1}{6\eta L}\right) D_x \le \frac{D_x}{2\eta L},\end{aligned}$$

where (i) uses assumption A1 and the definition of D_x, (ii) uses the fact that $\mu \le L$ and the condition $b/m_0 \le \Gamma \le 1/4$, and (iii) uses the fact that $\Gamma \le 1/4$, and thus $\eta L \le (1-\Gamma)/2 \le 3/8$. ∎

Proof of Theorem 4.4. Let

$$T^{(1)}(\epsilon) = \min\left\{T : \Lambda_T \ge \frac{D_x}{\epsilon \eta L}\right\}, \quad T^{(2)}(\epsilon) = \min\left\{T : \tilde{\Lambda}_T \ge \frac{D_H T_n^*}{\epsilon}\right\}. \tag{4.21}$$

Then for any $T \ge \max\{T^{(1)}(\epsilon), T^{(2)}(\epsilon)\}$,

$$\mathbb{E}(f(\tilde{x}_T) - f(x^*)) \le \frac{\epsilon}{2} + \frac{2\eta L m_0}{\Gamma b B_0}\epsilon \le \left(\frac{1}{2} + \Gamma\right)\epsilon \le \epsilon.$$

This entails

$$T(\epsilon) \le \max\{T^{(1)}(\epsilon), T^{(2)}(\epsilon)\}. \tag{4.22}$$

By definition, when $T \le T_\kappa^*$,

$$\Lambda_T \ge \alpha^T,$$

and when $T > T_\kappa^*$, since $\xi < 1$,

$$\Lambda_T = \alpha^{\lfloor T_\kappa^* \rfloor}\alpha^{(T - \lfloor T_\kappa^* \rfloor)/\xi} \ge \alpha^{T_\kappa^*}\alpha^{(T-T_\kappa^*)/\xi} = \frac{\kappa}{\eta L}\alpha^{(T-T_\kappa^*)/\xi}.$$

As a result,

$$T_1(\epsilon) \le \min\left\{\frac{\log\left(\frac{D_x}{\epsilon\eta L}\right)}{\log\alpha}, T_\kappa^* + \xi\frac{\log\left(\frac{D_x}{\epsilon\kappa}\right)}{\log\alpha}\right\} \tag{4.23}$$

and

$$\alpha^{T_1(\epsilon)} \le \min\left\{\frac{D_x}{\epsilon\eta L}, \frac{\kappa}{\eta L}\left(\frac{D_x}{\epsilon\kappa}\right)_*^{\xi}\right\}.$$

Similarly, when $T \le T_n^* \vee T_\kappa^*$,

$$\tilde{\Lambda}_T = \alpha^T,$$

and when $T > T_n^* \vee T_\kappa^*$,

$$\tilde{\Lambda}_T \ge \alpha^{T_n^* \vee T_\kappa^*} \alpha^{(T - T_n^* \vee T_\kappa^*)/\xi} \ge \alpha^{T_n^* + T_\kappa^*} \alpha^{(T - T_n^* - T_\kappa^*)/\xi}.$$

Thus,

$$\tilde{\Lambda}_T \ge \left(\sqrt{\frac{n}{B_0}} + \frac{1}{\eta\mu}\right)\alpha^{(T - T_n^* - T_\kappa^*)/\xi} \ge \frac{\tilde{\kappa}}{\sqrt{B_0} + \eta L}\alpha^{(T - T_n^* - T_\kappa^*)/\xi}.$$

As a result,

$$T_2(\epsilon) \le \min\left\{\frac{\log\left(\frac{D_H T_n^*}{\epsilon}\right)}{\log\alpha}, T_n^* + T_\kappa^* + \xi\frac{\log\left(\frac{D_H T_n^*(\sqrt{B_0}+\eta L)}{\epsilon\tilde{\kappa}}\right)}{\log\alpha}\right\}$$

and

$$\alpha^{T_2(\epsilon)} \le \min\left\{\frac{D_H T_n^*}{\epsilon}, \tilde{\kappa}\left(\frac{D_H T_n^*(\sqrt{B_0}+\eta L)}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}.$$

In summary, by (4.23)

$$\begin{aligned}
\alpha^{T(\epsilon)} &\le \alpha^{T_1(\epsilon)} + \alpha^{T_2(\epsilon)} \\
&\le \min\left\{\frac{D_x}{\epsilon\eta L}, \frac{\kappa}{\eta L}\left(\frac{D_x}{\epsilon\kappa}\right)_*^{\xi}\right\} + \min\left\{\frac{D_H T_n^*}{\epsilon}, \tilde{\kappa}\left(\frac{D_H T_n^*(\sqrt{B_0}+\eta L)}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\} \\
&= O\left(\min\left\{\frac{D_x + D_H T_n^*}{\epsilon}, \kappa\left(\frac{D_x}{\epsilon\kappa}\right)_*^{\xi} + \frac{D_H T_n^*}{\epsilon}, \tilde{\kappa}\left(\frac{D_x + D_H T_n^*}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}\right),
\end{aligned}$$

where the last line uses the monotonicity of the mapping $x \mapsto x^{1-\xi}$. By definition, $T_n^* = O(\log n) = \tilde{O}(1)$. As a result, $D_x + D_H T_n^* = \tilde{O}(\max\{D_x, D_H\}) = \tilde{O}(D)$, and thus

$$\alpha^{T(\epsilon)} = \tilde{O}\left(\min\left\{\frac{D}{\epsilon}, \kappa\left(\frac{D_x}{\epsilon\kappa}\right)_*^{\xi} + \frac{D_H}{\epsilon}, \tilde{\kappa}\left(\frac{D}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}\right).$$

The proof is then completed by replacing $\alpha^{T(\epsilon)}$ by $A(\epsilon)$ and (4.3). ∎

5. Mirror-proximal SCSG for composite problems. In this section we extend SCSG to composite problems in non-Euclidean spaces. Throughout this section we deal with problem (1.1), with $\mathcal{X}$ assumed to be a *subset* of a Hilbert space $\mathcal{X}_0$, equipped with an inner product $\langle\cdot,\cdot\rangle$. Let $\|\cdot\|_2$ denote the norm induced by the inner product, i.e., $\|x\|_2 = \sqrt{\langle x, x\rangle}$. For any convex function g, let g^* denote the convex conjugate of g,

$$g^*(x) = \sup_{y\in\mathcal{X}_0} \langle x, y\rangle - g(x).$$

For any differential convex function w, let $B_w(\cdot,\cdot)$ denote the Bregman divergence,

$$B_w(x,y) = w(x) - w(y) - \langle \nabla w(y), x - y \rangle .$$

We denote by $\mathbb{R}^+$ the set of nonnegative reals.

We define *mirror-proximal SCSG* as a variant of Algorithm 3.1 designed for composite problems (with $\psi \neq 0$). The algorithm is detailed below. The only difference lies in line 9 where the gradient step is replaced by a mirror-proximal step. This is the standard extension to composite problems in general Hilbert spaces (see, e.g., [13, 23, 3]). Note that when $\psi(x) \equiv 0$ and $w(x) = \|x\|_2^2/2$, Algorithm 5.1 reduces to Algorithm 3.1. Whenever $w(x) = \|x\|_2^2/2$, line 9 reduces to the proximal gradient step.

Unlike most of the literature on composite mirror-descent algorithms, our analysis requires a weaker condition on the distance-generating function $w(x)$. To state the condition, we define a class of functions which we refer to as *convex sup-homogeneous envelope functions (CHEFs)*.

Algorithm 5.1. Mirror-proximal SCSG for regularized finite-sum optimization.

Inputs: Number of stages T, initial iterate $\tilde{x}_0$, stepsizes $(\eta_j)_{j=1}^T$, block sizes $(B_j)_{j=1}^T$, inner loop sizes $(m_j)_{j=1}^T$, mini-batch sizes $(b_j)_{j=1}^T$.

Procedure

1: **for** $j = 1, 2, \ldots, T$ **do**
2: Uniformly sample a batch $\mathcal{I}_j \subset \{1, \ldots, n\}$ with $|\mathcal{I}_j| = B_j$;
3: $\mu_j \leftarrow \nabla f_{\mathcal{I}_j}(\tilde{x}_{j-1})$;
4: $x_0 \leftarrow \tilde{x}_{j-1}$;
5: Generate $N_j \sim \mathrm{Geom}\left(\frac{m_j}{m_j + b_j}\right)$;
6: **for** $k = 1, 2, \ldots, N_j$ **do**
7: Uniformly sample a batch $\tilde{\mathcal{I}}_{k-1}^{(j)} \subset \{1, \ldots, n\}$ with $|\tilde{\mathcal{I}}_{k-1}^{(j)}| = b_j$;
8: $\nu_{k-1}^{(j)} \leftarrow \nabla f_{\tilde{\mathcal{I}}_{k-1}^{(j)}}(x_{k-1}^{(j)}) - \nabla f_{\tilde{\mathcal{I}}_{k-1}^{(j)}}(x_0^{(j)}) + \mu_j$;
9: $x_k^{(j)} \leftarrow \arg\min_{x \in \mathcal{X}} \left(\left\langle \nu_{k-1}^{(j)}, y \right\rangle + \psi(y) + \frac{1}{\eta_j} B_w(y, x_{k-1}^{(j)}) \right)$;
10: **end for**
11: $\tilde{x}_j \leftarrow x_{N_j}$;
12: **end for**

Output: $\tilde{x}_T$.

Definition 5.1. *Given any increasing function $g : \mathbb{R}^+ \mapsto \mathbb{R}^+$ such that*

$$\lim_{\lambda \to 0} g^{-1}(\lambda) = 0,$$

a function $G : \mathcal{X}_0 \mapsto \mathbb{R}^+$ is a CHEF with parameters $(g(\cdot), C_G)$ if the following hold:

C1. *G is nonnegative with $G(0) = 0$, convex, and symmetric in the sense that $G(w) = G(-w)$ for any $w \in \mathcal{X}_0$;*

C2. *for any $w \in \mathcal{X}_0$ and $\lambda > 0$,*

$$G(\lambda w) \geq \lambda g(\lambda) G(w);$$

C3. *G^*, the convex conjugate of G, satisfies a* generalized Nemirovsky inequality *in the sense that for any set of independent mean-zero $\mathcal{X}_0$-valued random*

vectors $Z_1, \ldots, Z_m$,

$$\mathbb{E}G^*\left(\sum_{j=1}^m Z_j\right) \le C_G \sum_{j=1}^m \mathbb{E}G^*(Z_j).$$

Note that G^* is nonnegative, since for any $w \in \mathcal{X}_0$,

$$G^*(w) = \sup_{x \in \mathcal{X}_0} \langle x, w\rangle - G(x) \ge \langle 0, w\rangle - G(0) = 0.$$

Our first condition is imposed on the Bregman divergence induced by $w(x)$.

B1. There exists a CHEF such that for any $x, y \in \mathcal{X}$,

$$B_w(x, y) \ge G(x - y).$$

In the literature (see, e.g., [8, 13, 3]), it is common to consider the special case

$$G(w) = G_2(w; \|\cdot\|) = \frac{1}{2}\|w\|^2,$$

where $\|\cdot\|$ can be any norm, not necessarily $\|\cdot\|_2$, on $\mathcal{X}_0$. Srebro, Sridharan, and Tewani [49] considered a more general class of G's in the form of

$$G(w) = G_q(w; \|\cdot\|) = \frac{1}{q}\|w\|^q.$$

It is clear that G_q satisfies C1 and C2 for any $q > 1$ with $g(\lambda) = \lambda^{q-1}$.

To see that $G_q(w; \|\cdot\|)$ satisfies C3, we first consider the case where $q = 2$, where $\mathcal{X}_0 = \mathbb{R}^d$ is the Euclidean space and where $\|\cdot\| = \|\cdot\|_r$ for some $r \ge 1$, with

$$\|x\|_r = \begin{cases} \left(\sum_{i=1}^n |x_i|^r\right)^{1/r} & (1 \le r < \infty), \\ \max_{i=1}^n |x_i|, & (r = \infty). \end{cases}$$

Then $\|\cdot\|_* = \|\cdot\|_{r'}$ where $r' = r/(r-1)$. By Lemma C.3 of [26],

$$G_2^*(x; \|\cdot\|_r) = \frac{1}{2}\|x\|_{r'}^2.$$

By Nemirovsky's inequality (Theorem 2.2 of [14]), for any independent mean-zero random vectors $Z_1, \ldots, Z_m \in \mathbb{R}^d$,

$$\mathbb{E}\left\|\sum_{j=1}^n Z_j\right\|_{r'}^2 \le K_{\mathrm{Nem}}(d, r') \sum_{j=1}^n \mathbb{E}\|Z_j\|_{r'}^2,$$

where

$$K_{\mathrm{Nem}}(d, r') \le \min\{r' - 1, 2e \log d\}.$$

Thus, whenever $r' = O(1)$, $K_{\mathrm{Nem}}(d, r') = O(1)$. Even when $r' = \infty$, in which case $r = 1$, $K_{\mathrm{Nem}}(d, r')$ scales as $\log d$.

Generally, given $G(x) = \|x\|^q/q$ for $q \in (1, 2)$ and a norm $\|\cdot\|$ on a general Hilbert space $\mathcal{X}$, Lemma C.3 implies that $G^*(x) = \|x\|_*^p/p$ where $p = q/(q-1)$. Then property C3 is equivalent to the condition that $\mathcal{X}_0$ has *Martingale type p* (see, e.g., [40]). In particular, when $\mathcal{X}_0 = \mathbb{R}^d$ and $\|\cdot\| = \|\cdot\|_r$ with $r \le q$, we prove in Proposition C.5 of

[26] that $G^*(x)$ satisfies the property C3, using Hanner's inequality [16]. In summary, the property C3 is satisfied in almost all cases that have been commonly studied in the literature on mirror-descent methods.

Besides assumption B1 we need the analogous assumptions A1 and A2 for the smoothness and strong convexity of the objectives.

B2. $0 \le B_{f_i}(x,y) \le LG(x-y)$ for all i and $x, y \in \mathcal{X}$.

B3. $F(x) - F(x^*) \ge \mu B_w(x^*, x)$ for some $\mu \ge 0$.

It is easy to see that assumptions B2 and B3 reduce to A1 and A2 when $\mathcal{X} = \mathbb{R}^d, G(x) = w(x) = \|x\|_2^2/2$. Note that B3 only requires strong convexity at x^*; it does not requires global strong convexity.

Finally, we modify the definitions of D_x, D_H as

$$D_x = L\mathbb{E}B_w(x^*, \tilde{x}_0), \quad D_H = \frac{L}{n}\sum_{i=1}^{n} G^*\left(\frac{\nabla f_i(x^*) - \nabla f(x^*)}{L}\right),$$

where x^* is the optimum of F. It is straightforward to show that D_x and D_H coincide with (2.1) up to a constant two when $\mathcal{X}_0 = \mathcal{X} = \mathbb{R}^d$ and $w(x) = \|x\|_2^2/2$. We also define an extra quantity D_F as

$$D_F = \mathbb{E}[F(\tilde{x}_0) - F(x^*)].$$

In the unregularized case, assumptions B1 and B2 imply that $D_F \le D_x$. However, this comparison may not hold in the regularized case. Finally, we redefine D as the maximum of D_x, D_H, and D_F.

5.1. Main results. Similarly to the unregularized case, we present results on the one-epoch analysis, the multiple-epoch analysis, and the complexity analysis. The results are almost the same as those in section 4, though the proofs are much more involved. All proofs are relegated to Appendix B in [26].

THEOREM 5.2. *Fix any $\xi > 0$ and $\Gamma = 1/4\alpha^{1/\xi}$. Assume that*

$$6\eta_j L \le \min\left\{1, b_j g\left(\frac{\Gamma}{3C_G}\right)\right\}, \quad m_j \ge \max\{b_j, 4C_G\}/\Gamma,$$

and

$$B_j \ge \frac{5}{8\Gamma} g\left(\frac{\Gamma}{3C_G}\right)\frac{m_j}{g(1/m_j)}.$$

Then under assumptions B1–B3*,*

$$\begin{aligned}&\mathbb{E}(F(\tilde{x}_j) - F(x^*)) + \left(\frac{4b_j(1-\Gamma)}{3\eta}\frac{1}{m_j} + \frac{\mu}{3}\right)\mathbb{E}B_w(x^*, \tilde{x}_j)\\ &\le 4\Gamma\mathbb{E}(F(\tilde{x}_{j-1}) - F(x^*)) + \frac{4b_j(1-\Gamma)}{3\eta}\frac{1}{m_j}\mathbb{E}B_w(x^*, \tilde{x}_{j-1}) + \frac{D_H}{6\alpha^j}I(B_j < n).\end{aligned}$$

Theorems 5.2 and 4.1 give almost the same result, up to constants, except that Theorem 4.1 has an additional term $\frac{\eta L}{b}$ in the coefficient of D_H. In the cases where η is small and b is large, Theorem 4.1 gives a better guarantee. However, in our settings for SCSG, both ηL and b are taken as $O(1)$, and thus the theorems yield the same results up to the constant.

To set the parameters for SCSG in this general case, we still take a constant stepsize, a constant mini-batch size, and a geometrically increasing sequence for m_j.

In contrast, B_j should scale as $m_j/g(1/m_j)$. This coincides with Theorem 4.2 since $g(x) = x$ in the unregularized case with the usual strong convexity condition (assumption A2).

THEOREM 5.3. *Fix any given constant $\alpha > 1$, $m_0 > 0$, and $\xi \in (0,1)$. Let*

$$\eta_j \equiv \eta, \quad b_j \equiv b, \quad m_j = m_0\alpha^j, \quad B_j = \left\lceil \frac{5\alpha^{1/\xi}}{2} g\left(\frac{1}{12C_G\alpha^{1/\xi}}\right) \frac{m_j}{g(1/m_j)} \right\rceil.$$

Assume that

$$m_0 \geq 4\alpha^{1/\xi}\max\{b, 4C_G\}, \quad 6\eta L \leq \min\left\{1, g\left(\frac{1}{12C_G\alpha^{1/\xi}}\right) b\right\}.$$

Then under assumptions B1–B3*,*

$$\mathbb{E}(F(\tilde{x}_j) - F(x^*)) \leq \Lambda_T^{-1}\left(D_F + \frac{D_x}{3\eta L}\right) + \tilde{\Lambda}_T^{-1}\frac{D_H}{6}(T \wedge T_n^*),$$

where Λ_T and $\tilde{\Lambda}_T$ are defined as in Theorem 4.2*.*

Theorem 5.3 gives almost the same result as Theorem 4.2, except that the second term is loose up to a term $\frac{\eta L}{b}$. As mentioned before, this is a constant in our setting, and thus the gap is negligible in terms of the theoretical complexity.

Applying the same argument as in Theorem 4.4, we can derive the theoretical complexity. Again it coincides with Theorem 4.4 in the unregularized case with the usual strong convexity condition (i.e., assumption A2).

THEOREM 5.4. *Under the specification of Theorem* 5.3*, we have*

$$\mathbb{E}C_{\text{comp}}(\epsilon) = O\left(\frac{A(\epsilon)}{g(1/A(\epsilon))} \wedge (A(\epsilon) + n\log A(\epsilon))\right),$$

where $\tilde{\kappa} = \alpha^{T_n^*} + \kappa$ and

$$A(\epsilon) = \tilde{O}\left(\min\left\{\frac{D}{\epsilon}, \kappa\left(\frac{D_x + D_F}{\epsilon\kappa}\right)_*^{\xi} + \frac{D_H}{\epsilon}, \tilde{\kappa}\left(\frac{D}{\epsilon\tilde{\kappa}}\right)_*^{\xi}\right\}\right).$$

Interestingly, in the uniformly convex case [21], where

$$B_w(x,y) \geq G(x-y) = \frac{1}{q}\|x-y\|^q,$$

we can set $g(\lambda) = \lambda^{q-1}$. Then Theorem 5.4 implies that

$$\mathbb{E}C_{\text{comp}}(\epsilon) = O\left(A(\epsilon)^q \wedge (A(\epsilon) + n\log A(\epsilon))\right). \tag{5.1}$$

Recalling that $\tilde{\tilde{O}}$ hides the negligible terms $\left(\frac{D}{\epsilon\kappa}\right)^{2\xi}$ and $\tilde{\kappa} = \alpha^{T_n^*} + \kappa = O(n+\kappa)$, we can rewrite (5.1) as

$$\mathbb{E}C_{\text{comp}}(\epsilon) = \tilde{\tilde{O}}\left(\left(\frac{D}{\epsilon}\right)^q \wedge \left(\kappa^q + \left(\frac{D_H}{\epsilon}\right)^q\right) \wedge \left(n + \frac{D}{\epsilon}\right) \wedge (n+\kappa)\right).$$

The first term matches the bound in [49]. However, to the best of our knowledge, the other terms have not been investigated in the literature.

6. Conclusions. We have presented SCSG, a gradient-based algorithm for the convex finite-sum optimization problem, which is ϵ-independent and almost universal. These properties arise from two ideas: *geometrization* and *batching variance reduction*. SCSG achieves strong adaptivity to both the target accuracy and to strong convexity with complexity

$$\tilde{\tilde{O}}\left(\frac{D^2}{\epsilon^2} \wedge \left(\kappa^2 + \frac{D_H^2}{\epsilon^2}\right) \wedge \left(n + \frac{D}{\epsilon}\right) \wedge (n+\kappa)\right)$$

up to negligible terms. This is strictly better than other existing adaptive algorithms. We also present a mirror-proximal version of SCSG for problems involving non-Euclidean geometry. Our analysis requires the Bregman divergence to be lower bounded by a CHEF, a construct which unifies and generalizes existing work on mirror-descent methods. We derive a set of technical tools to deal with CHEFs which may be of interest in other problems.

A major direction for further research is delineating optimal rates for algorithms that exhibit adaptivity. Our conjecture is that the optimal complexity for a reasonably large class of algorithms that do not require knowledge of μ is

$$\tilde{O}\left(\left(\sqrt{\frac{D_x}{\epsilon}} + \frac{D_H^2}{\epsilon^2}\right) \wedge \frac{\kappa D_H}{\epsilon} \wedge \left(n + \sqrt{\frac{nD_x}{\epsilon}}\right) \wedge (n+\kappa)\right).$$

We believe that momentum terms are required to achieve such a rate.

REFERENCES

[1] A. Agarwal and L. Bottou, *A Lower Bound for the Optimization of Finite Sums*, preprint, https://arxiv.org/abs/1410.0723, 2014.

[2] A. Agarwal, M. J. Wainwright, P. L. Bartlett, and P. K. Ravikumar, *Information-theoretic lower bounds on the oracle complexity of convex optimization*, in Advances in Neural Information Processing Systems 22, Y. Bengio et al., eds., Curran Associates, 2009, pp. 1–9.

[3] Z. Allen-Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, in Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, ACM, 2017, pp. 1200–1205.

[4] Z. Allen-Zhu and E. Hazan, *Optimal black-box reductions between optimization objectives*, in Advances in Neural Information Processing Systems 29, D. D. Lee et al., eds., Curran Associates, 2016, pp. 1614–1622.

[5] Z. Allen-Zhu and Y. Yuan, *Improved SVRG for non-strongly-convex or sum-of-non-convex objectives*, in Proceedings of the 33rd International Conference on Machine Learning, M. F. Balcan and K. Q. Weinberger eds., PMLR, 2016, pp. 1080–1089.

[6] Y. Arjevani, *Limitations on Variance-Reduction and Acceleration Schemes for Finite Sum Optimization*, preprint, https://arxiv.org/abs/1706.01686, 2017.

[7] F. Bach and E. Moulines, *Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$*, in Advances in Neural Information Processing Systems 26, C. J. C. Burges et al., eds., Curran Associates, 2013, pp. 773–781.

[8] A. Beck and M. Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Oper. Res. Lett., 31 (2003), pp. 167–175.

[9] Z. Chen, Y. Xu, E. Chen, and T. Yang, *SADAGRAD: Strongly adaptive stochastic gradient methods*, in Proceedings of the 35th International Conference on Machine Learning, J. Dy and A. Krause, eds., PMLR, 2018, pp. 913–921.

[10] A. Defazio, F. Bach, and S. Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani et al., eds., Curran Associates, 2014, pp. 1646–1654.

[11] A. Dieuleveut, N. Flammarion, and F. Bach, *Harder, better, faster, stronger convergence rates for least-squares regression*, J. Mach. Learn. Res., 18 (2017), pp. 3520–3570.

[12] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res., 12 (2011), pp. 2121–2159.
[13] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, *Composite objective mirror descent*, in Proceedings of the 23rd Conference on Learning Theory, Haifa, Israel, 2010, pp. 14–26.
[14] L. Dümbgen, S. A. Van De Geer, M. C. Veraar, and J. A. Wellner, *Nemirovski's inequalities revisited*, Amer. Math. Monthly, 117 (2010), pp. 138–160.
[15] N. Flammarion and F. Bach, *From averaging to acceleration, there is only a step-size*, in Proceedings of the 28th Conference on Learning Theory, P. Grünwald, E. Hazan, and S. Kale, eds., PMLR, 2015, pp. 658–695.
[16] O. Hanner, *On the uniform convexity of L_p and ℓ_p*, Ark. Mat., 3 (1956), pp. 239–244.
[17] E. Hazan and S. Kakade, *Revisiting the Polyak Step Size*, preprint, https://arxiv.org/abs/1905.00313, 2019.
[18] E. Hazan and S. Kale, *An Optimal Algorithm for Stochastic Strongly-Convex Optimization*, preprint https://arxiv.org/abs/1006.2425, 2010.
[19] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems 26, C. J. C. Burges et al., eds., Curran Associates, 2013, pp. 315–323.
[20] A. Juditsky, A. Nemirovski, and C. Tauvel, *Solving variational inequalities with stochastic mirror-prox algorithm*, Stochastic Systems, 1 (2011), pp. 17–58.
[21] A. Juditsky and Y. Nesterov, *Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization*, Stochastic Systems, 4 (2014), pp. 44–80.
[22] G. Lan, *An optimal method for stochastic composite optimization*, Math. Programming, 133 (2012), pp. 365–397.
[23] G. Lan, *An Optimal Randomized Incremental Gradient Method*, preprint, https://arxiv.org/abs/1507.02000, 2015.
[24] G. Lan, Z. Li, and Y. Zhou, *A Unified Variance-Reduced Accelerated Gradient Method for Convex Optimization*, preprint, https://arxiv.org/abs/1905.12412, 2019.
[25] L. Lei and M. I. Jordan, *Less than a Single Pass: Stochastically Controlled Stochastic Gradient Method*, preprint, https://arxiv.org/abs/1609.03261, 2016.
[26] L. Lei and M. I. Jordan, *On the Adaptivity of Stochastic Gradient-Based Optimization*, preprint, https://arxiv.org/abs/1904.04480, 2019.
[27] L. Lei, C. Ju, J. Chen, and M. I. Jordan, *Non-convex finite-sum optimization via SCSG methods*, in Advances in Neural Information Processing Systems 30, I. Guyon et al., eds., Curran Associates, 2017, pp. 2348–2358.
[28] Y. K. Levy, A. Yurtsever, and V. Cevher, *Online adaptive methods, universality and acceleration*, in Advances in Neural Information Processing Systems 31, S. Bengio et al., eds., Curran Associates, 2018, pp. 6500–6509.
[29] M. Li, T. Zhang, Y. Chen, and A. J. Smola, *Efficient mini-batch training for stochastic optimization*, in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 661–670.
[30] H. Lin, J. Mairal, and Z. Harchaoui, *A universal catalyst for first-order optimization*, in Advances in Neural Information Processing Systems 28, C. Cortes et al., eds., Curran Associates, 2015, pp. 3384–3392.
[31] Q. Lin, Z. Lu, and L. Xiao, *An accelerated proximal coordinate gradient method*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani et al., eds., Curran Associates, 2014, pp. 3059–3067.
[32] S. Ma, R. Bassily, and M. Belkin, *The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-Parametrized Learning*, preprint, https://arxiv.org/abs/1712.06559, 2017.
[33] E. Moulines and F. R. Bach, *Non-asymptotic analysis of stochastic approximation algorithms for machine learning*, in Advances in Neural Information Processing Systems 24, J. Shawe-Taylor et al., eds., Curran Associates, 2011, pp. 451–459.
[34] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, *Robust stochastic approximation approach to stochastic programming*, SIAM J. Optim., 19 (2009), pp. 1574–1609, https://doi.org/10.1137/070704277.
[35] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.
[36] Y. Nesterov, *Universal gradient methods for convex optimization problems*, Math. Programming, 152 (2015), pp. 381–404.
[37] L. M. Nguyen, M. van Dijk, D. T. Phan, P. H. Nguyen, T.-W. Weng, and J. R. Kalagnanam, *Finite-Sum Smooth Optimization with SARAH*, preprint, https://arxiv.org/abs/1901.07648v2, 2019.

[38] A. Nitanda, *Accelerated stochastic gradient descent for minimizing finite sums*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, A. Gretton and C. R. Robert, PMLR, 2016, pp. 195–203.
[39] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui, *Catalyst Acceleration for Gradient-Based Non-convex Optimization*, preprint, https://arxiv.org/abs/1703.10993, 2017.
[40] G. Pisier, *Martingales with values in uniformly convex spaces*, Israel J. Math., 20 (1975), pp. 326–350.
[41] B. T. Polyak, *A new method of stochastic approximation type,* Avtomat. i Telemekh., 1990, no. 7, pp. 98–107 (in Russian); translation in Automat. Remote Control, 51 (1990), part 2, pp. 937–946.
[42] B. T. Polyak and A. B. Juditsky, *Acceleration of stochastic approximation by averaging*, SIAM J. Control Optim., 30 (1992), pp. 838–855, https://doi.org/10.1137/0330046.
[43] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, *Stochastic Variance Reduction for Nonconvex Optimization*, preprint, https://arxiv.org/abs/1603.06160, 2016.
[44] N. L. Roux, M. Schmidt, and F. Bach, *A stochastic gradient method with an exponential convergence rate for finite training sets*, in Advances in Neural Information Processing Systems 25, F. Pereira et al., eds., Curran Associates, 2012, pp. 2663–2671.
[45] D. Ruppert, *Efficient Estimators from a Slowly Convergent Robbins-Monro Procedure*, Tech. report 781, School of Oper. Res. and Ind. Eng., Cornell University, 1988.
[46] S. Shalev-Shwartz and T. Zhang, *Proximal Stochastic Dual Coordinate Ascent*, preprint, https://arxiv.org/abs/1211.2717, 2012.
[47] S. Shalev-Shwartz and T. Zhang, *Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization*, in Math. Program., 155 (2016), Ser. A, pp. 105–145.
[48] O. Shamir, *Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization*, preprint, https://arxiv.org/abs/1109.5647v1, 2011.
[49] N. Srebro, K. Sridharan, and A. Tewari, *On the universality of online mirror descent*, in Advances in Neural Information Processing Systems 24, J. Shawe-Taylor et al., eds., Curran Associates, 2011, pp. 2645–2653.
[50] S. Vaswani, F. Bach, and M. Schmidt, *Fast and Faster Convergence of SGD for Over-Parameterized Models and an Accelerated Perceptron*, preprint, https://arxiv.org/abs/1810.07288, 2018.
[51] S. Vaswani, A. Mishkin, I. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien, *Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates*, preprint, https://arxiv.org/abs/1905.09997, 2019.
[52] B. Woodworth and N. Srebro, *Tight Complexity Bounds for Optimizing Composite Objectives*, preprint, https://arxiv.org/abs/1605.08003, 2016.
[53] Y. Xu, Q. Lin, and T. Yang, *Adaptive SVRG methods under error bound conditions with unknown growth parameter*, in Advances in Neural Information Processing Systems 30, I. Guyon et al., eds., Curran Associates, 2017, pp. 3279–3289.
[54] Y. Xu, Q. Lin, and T. Yang, *Accelerate stochastic subgradient method by leveraging local growth condition*, Anal. Appl., 17 (2019), pp. 773–818.