SGD: General Analysis and Improved Rates
Robert Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, Peter Richtárik

To cite this version:
Robert Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, et al.. SGD: General Analysis and Improved Rates. International Conference on Machine Learning, Jun 2019, Los Angeles, United States. hal-02365318

HAL Id: hal-02365318
https://hal.telecom-paris.fr/hal-02365318
Submitted on 15 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
SGD: General Analysis and Improved Rates

Robert M. Gower 1  Nicolas Loizou 2  Xun Qian 3  Alibek Sailanbayev 3  Egor Shulgin 4  Peter Richtárik 3 2 4

Abstract

We propose a general yet simple theorem describing the convergence of SGD under the arbitrary sampling paradigm. Our theorem describes the convergence of an infinite array of variants of SGD, each of which is associated with a specific probability law governing the data selection rule used to form minibatches. This is the first time such an analysis is performed, and most of our variants of SGD were never explicitly considered in the literature before. Our analysis relies on the recently introduced notion of expected smoothness and does not rely on a uniform bound on the variance of the stochastic gradients. By specializing our theorem to different mini-batching strategies, such as sampling with replacement and independent sampling, we derive exact expressions for the stepsize as a function of the mini-batch size. With this we can also determine the mini-batch size that optimizes the total complexity, and show explicitly that as the variance of the stochastic gradient evaluated at the minimum grows, so does the optimal mini-batch size. For zero variance, the optimal mini-batch size is one. Moreover, we prove insightful stepsize-switching rules which describe when one should switch from a constant to a decreasing stepsize regime.

1. Introduction

We consider the optimization problem

\[ x^* = \arg \min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x), \]

where each \( f_i : \mathbb{R}^d \to \mathbb{R} \) is smooth (but not necessarily convex). Further, we assume that \( f \) has a unique global minimizer \( x^* \) and is \( \mu \)-strongly quasi-convex (Karimi et al., 2016; Necoara et al., 2018):

\[ f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle + \frac{\mu}{2} \| x^* - x \|^2 \]  

for all \( x \in \mathbb{R}^d \).

1.1. Background and contributions

Stochastic gradient descent (SGD) (Robbins & Monro, 1951; Nemirovski & Yudin, 1978; 1983; Shalev-Shwartz et al., 2007; Nemirovski et al., 2009; Hardt et al., 2016), has become the workhorse for training supervised machine learning problems which have the generic form (1).

Linear convergence of SGD. Moulines & Bach (2011) provided a non-asymptotic analyses of SGD showing linear convergence for strongly convex \( f \) up to a certain noise level. Needell et al. (2016) improved upon these results by removing the quadratic dependency on the condition number in the iteration complexity results, and considered importance sampling. The analysis of Needell et al. (2016) was later extended to a mini-batch variant where the mini-batches are formed by partitioning the data (Needell & Ward, 2017). These works are the main starting point for ours.

Contributions: We further tighten and generalize these results to virtually all forms of sampling. We introduce an expected smoothness assumption (Assumption 2.1), first introduced in (Gower et al., 2018) in the context of a certain class of variance-reduced methods. This assumption is a joint property of \( f \) and the sampling scheme \( D \) utilized by an SGD method, and allows us prove a generic complexity result (Theorem 3.1) that holds for arbitrary sampling schemes \( D \). Our work is the first time SGD is analysed under this assumption.

Gradient noise assumptions. Shamir & Zhang (2013) extended the analysis of SGD to convex non-smooth optimiza-

\[ 1 \text{This assumption can be relaxed; but for simplicity of exposition we enforce it.} \]
tion (including the strongly convex case). However, their proofs still rely on the assumption that the variance of the stochastic gradient is bounded for all iterates of the algorithm: there exists $c \in \mathbb{R}$ such that $E_k[\|\nabla f_i(x^k)\|^2] \leq c$ for all $k$. The same assumption was used in the analysis of several recent papers ([Recht et al., 2011], [Hazan & Kale, 2014], [Rakhlin et al., 2012]). A much more relaxed weak growth assumption $E_k[\|\nabla f_i(x^k)\|^2] \leq c_1 + c_2 E_k[\|\nabla f(x^k)\|^2]$ for all $k$, was apparently first used in the later 90’s to prove the asymptotic convergence of SGD (see Proposition 4.2 of Bertsekas & Tsitsiklis [1996]). Bottou et al. (2018) established a linear convergence of SGD under this weak growth assumption. Recently, Nguyen et al. (2018) turn this assumption into a theorem by establishing formulas $c_1$ and $c_2$ under some reasonable conditions, and provide further insights into the workings of SGD and its parallel asynchronous cousin, Hogwild!.

Contributions: Our analysis does not directly assume a growth condition. Instead, we make use of the remarkably weak expected smoothness assumption.

Optimal mini-batch size. Recently it was experimentally shown by Goyal et al. (2017) that using larger mini-batches is key to efficient training of large scale non-convex problems, leading to the training of ImageNet in under 1 hour. The authors conjectured that the stepsize should grow linearly with the mini-batch size.

Contributions: We prove (see Section 4.1) that this is the case, up to a certain optimal mini-batch size, and provide exact formulas for the dependency of the stepsizes on the mini-batch sizes.

Learning schedules. Chee & Toulis (2018) develop techniques for detecting the convergence of SGD within a region around the solution.

Contributions: We provide a closed-form formula for when SGD switch from a constant stepsize to a decreasing stepsize (see Theorem 3.2). Further, we clearly show how the optimal stepsize (learning rate) increases and the iteration complexity decreases as the mini-batch size increases for both independent sampling and sampling with replacement. We also recover the well known $L/\mu \log(1/\epsilon)$ convergence rate of gradient descent (GD) when the mini-batch size is $\alpha$; this is the first time a generic SGD analysis recovers the correct rate of GD.

Over-parametrized models. There has been some recent work in analysing SGD in the setting where the underlying model being trained has more parameters than there is data available. In this zero-noise setting, Ma et al. (2018) showed that SGD converges linearly.

Contributions: In the case of over-parametrized models, we extend the findings of Ma et al. (2018) to independent sampling and sampling with replacement by showing that the optimal mini-batch size is 1. Moreover, we provide results in the more general setting where the model is not necessarily over-parametrized.

Practical performance. We corroborate our theoretical results with extensive experimental testing.

1.2. Stochastic reformulation

In this work we provide a single theorem through which we can analyse all importance sampling and mini-batch variants of SGD. To do this, we need to introduce a sampling vector which we will use to re-write our problem (1).

Definition 1.1. We say that a random vector $v \in \mathbb{R}^n$ drawn from some distribution $\mathcal{D}$ is a sampling vector if its mean is the vector of all ones:

$$E_{\mathcal{D}}[v_i] = 1, \quad \forall i \in [n].$$

(3)

With each distribution $\mathcal{D}$ we now introduce a stochastic reformulation of (1) as follows

$$\min_{x \in \mathbb{R}^d} E_{\mathcal{D}}[f_i(x) := \frac{1}{n} \sum_{i=1}^{n} v_i f_i(x)].$$

(4)

By the definition of the sampling vector, $f_i(x)$ and $\nabla f_i(x)$ are unbiased estimators of $f(x)$ and $\nabla f(x)$, respectively, and hence problem (4) is indeed equivalent (i.e., a reformulation) of the original problem (1). In the case of the gradient, for instance, we get

$$E_{\mathcal{D}}[\nabla f_i(x)] = \frac{1}{n} \sum_{i=1}^{n} E_{\mathcal{D}}[v_i] \nabla f_i(x) = E_{\mathcal{D}}[v] \nabla f(x).$$

(5)

Similar but different stochastic reformulations were recently proposed by Richtárik & Takáč (2017) and further used in Loizou & Richtárik (2017), 2019 for the more special problem of solving linear systems, and by Gower et al. (2018) in the context of variance-reduced methods. Reformulation (4) can be solved using SGD in a natural way:

$$x^{k+1} = x^k - \gamma^k \nabla f_i(x^k)$$

(6)

where $\nu^k \sim \mathcal{D}$ is sampled i.i.d. at each iteration and $\gamma^k > 0$ is a stepsize. However, for different distributions $\mathcal{D}$, (6) has a different interpretation as an SGD method for solving the original problem (1). In our main result we will analyse (6) for any $\mathcal{D}$ satisfying (5). By substituting specific choices of $\mathcal{D}$, we obtain specific variants of SGD for solving (1).
2. Expected Smoothness and Gradient Noise

In our analysis of SGD \cite{bottou2018optimization} applied to the stochastic reformulation \cite{karimi2016linear}, we rely on a generic and remarkably weak assumption of expected smoothness, which we now define and relate to existing growth conditions.

2.1. Expected smoothness

Expected smoothness \cite{gower2018generalization} is an assumption that combines both the properties of the distribution \( \mathcal{D} \) and the smoothness properties of function \( f \).

Assumption 2.1 (Expected Smoothness). We say that \( f \) is \( \mathcal{L} \)-smooth in expectation with respect to a distribution \( \mathcal{D} \) if there exists \( \mathcal{L} = \mathcal{L}(f, \mathcal{D}) > 0 \) such that

\[
\mathbb{E}_\mathcal{D} \left[ \| \nabla f_v(x) - \nabla f_v(x^*) \|^2 \right] \leq 2\mathcal{L}(f(x) - f(x^*)),
\]

for all \( x \in \mathbb{R}^d \). For simplicity, we will write \( f, \mathcal{D} \sim ES(\mathcal{L}) \) to say that \( \mathcal{L} \) holds. When \( \mathcal{D} \) is clear from the context, we will often ignore mentioning it, and simply state that the expected smoothness constant is \( \mathcal{L} \).

There are scenarios where the above inequality is tight. Indeed, in the setting of stochastic reformulations of linear systems considered in \cite{richtarik2017iteration}, one has \( f_v(x) = \frac{1}{2}\| \nabla f_v(x) \|^2, \nabla f_v(x^*) = 0 \) and \( f_v(x^*) = 0 \), which means that \( \mathcal{L} \) holds as an identity with \( \mathcal{L} = 1 \).

In Section 3.3 we show how convexity and \( L_i \)-smoothness of \( f_i \) implies expected smoothness. However, the opposite implication does not hold. Indeed, the expected smoothness assumption can hold even when the \( f_i \)'s and \( f \) are not convex, as we show in the next example.

Example 2.2 (Non-convex and expected smoothness). Let \( f_i = \phi \) for \( i = 1, \ldots, n \), where \( \phi \) is a \( L_\phi \)-smooth and non-convex function which has a global minimum \( x^* \in \mathbb{R}^d \) (such functions exist\footnote{There exists invex functions that satisfy these conditions \cite{karimi2016linear}. As an example \( \phi(x) = x^2 + 3 \sin^2(x) \) is smooth, non-convex, and has a unique global minimizer.}). Consequently \( f = \phi \) and \( f_v = \sum_i v_i \phi \). Letting \( \theta := \mathbb{E}_\mathcal{D} \left[ \left( \sum_i v_i \right)^2 \right] \), we have

\[
\mathbb{E}_\mathcal{D} \left[ \| \nabla f_v(x) - \nabla f_v(x^*) \|^2 \right] = \frac{\theta}{\pi^2} \| \nabla \phi(x) - \nabla \phi(x^*) \|^2 \leq \frac{2\mathcal{L}^2}{\pi^2}(f(x) - f(x^*)),
\]

where the last inequality follows from Proposition A.1. So, \( (f, \mathcal{D}) \sim ES(\mathcal{L}) \) for \( \mathcal{L} = \frac{\theta L_\phi}{\pi^2} \).

2.2. Gradient noise

Our second key assumption is finiteness of gradient noise, defined next:

Assumption 2.3 (Finite Gradient Noise). The gradient noise \( \sigma = \sigma(f, \mathcal{D}) \), defined as follows is finite

\[
\sigma^2 := \mathbb{E}_\mathcal{D}[\| \nabla f_v(x^*) \|^2].
\]

This is a very weak assumption, and should intuitively be seen as an assumption on \( \mathcal{D} \) rather than on \( f \). For instance, if the sampling vector \( \nu \) is non-negative with probability one and \( \mathbb{E}[v_i \sum_j v_j] \) is finite for all \( i \), then \( \sigma \) is finite. When \( \mathcal{D} \) is the training problem of an over-parametrized model, which often occurs in deep neural networks, each individual loss function \( f_i \) attains its minimum at \( x^* \), and thus \( \nabla f_i(x^*) = 0 \). It follows that \( \sigma = 0 \).

2.3. Key lemma and connection to the weak growth condition

A common assumption used to prove the convergence of SGD is uniform boundedness of the stochastic gradients\footnote{Or it is assumed that \( \mathbb{E}[\| \nabla f_v(x^k) \|^2] \leq c \) for all \( k \) iterates. But this too has issues since it implicitly assumes that the iterates remain within a compact set, and yet it used to prove the convergence to within a compact set, raising issues of a circular argument.}. For instance, if the sampling vector \( \nu \) is non-negative with probability one and \( \mathbb{E}[v_i \sum_j v_j] \) is finite for all \( i \), then \( \sigma \) is finite. When \( \mathcal{D} \) is the training problem of an over-parametrized model, which often occurs in deep neural networks, each individual loss function \( f_i \) attains its minimum at \( x^* \), and thus \( \nabla f_i(x^*) = 0 \). It follows that \( \sigma = 0 \).

Corollary 2.5. If \( (f, \mathcal{D}) \sim ES(\mathcal{L}) \) and if \( \sigma = 0 \), then \( f \) satisfies the weak growth condition

\[
\mathbb{E}_\mathcal{D}[\| \nabla f_v(x) \|^2] \leq 2\rho(f(x) - f(x^*)),
\]

with \( \rho = 2\mathcal{L} \).

This corollary should be contrasted with Proposition 2 in \cite{vaswani2018convergence} and Lemma 1 in \cite{nguyen2018finite}, where it is shown, by assuming the \( f_i \) functions to be smooth and convex, that the weak growth condition holds with \( \rho = 2L_{\max} \). However, as we will show in Lemma F.1, \( L_{\max} \geq \mathcal{L} \), and hence our bound is often tighter.
3. Convergence Analysis

3.1. Main results

We now present our main theorem.

**Theorem 3.1.** Assume $f$ is $\mu$-quasi-strongly convex and that $(f, D) \sim ES(\mathcal{L})$. Choose $\gamma^k = \gamma \in (0, 1/2\mu]$ for all $k$. Then iterates of SGD given by (6) satisfy:

$$E\|x^k - x^*\|^2 \leq (1 - \gamma\mu)^k \|x^0 - x^*\|^2 + 2\gamma^2 \frac{\mu}{\epsilon^2}.$$  

(10)

Hence, given any $\epsilon > 0$, choosing stepsize

$$\gamma = \min \left\{ \frac{1}{2\mu}, \frac{\mu}{4\epsilon^2} \right\},$$

(11)

and

$$k \geq \max \left\{ \frac{2\epsilon}{\mu}, \frac{2\gamma\sigma^2}{\epsilon^2} \right\} \log \left( \frac{2\|x^0 - x^*\|^2}{\epsilon} \right),$$

(12)

implies $E\|x^k - x^*\|^2 \leq \epsilon$.

Note that we do not assume $f_i$ nor $f$ to be convex. Theorem 3.1 states that SGD converges linearly up to the additive constant $2\gamma\sigma^2/\mu$ which depends on the gradient noise $\sigma^2$ and on the stepsize $\gamma$. We obtain a more accurate solution with a smaller stepsize, but then the convergence rate slows down. Since we control $D$, we also control $\sigma^2$ and $\mathcal{L}$ (we compute these parameters for several distributions $D$ in Section 3.3).

Furthermore, we can control this additive constant by carefully choosing the stepsize, as shown in the next result.

**Theorem 3.2 (Decreasing stepsizes).** Assume $f$ is $\mu$-quasi-strongly convex and that $(f, D) \sim ES(\mathcal{L})$. Let $\mathcal{K} := \mathcal{L}/\mu$ and

$$\gamma^k = \begin{cases} \frac{1}{2\mu} & \text{for } k \leq 4\lfloor \mathcal{K} \rfloor \\ \frac{2(1 + k)^{\mathcal{K}}}{(k + 1)^{\mathcal{K}}\mu} & \text{for } k > 4\lfloor \mathcal{K} \rfloor. \end{cases}$$

(13)

If $k \geq 4\lfloor \mathcal{K} \rfloor$, then SGD iterates given by (6) satisfy:

$$E\|x^k - x^*\|^2 \leq \frac{\sigma^2}{\mu^2 \epsilon^2} + \frac{16\lfloor \mathcal{K} \rfloor^2}{\epsilon^2} \|x^0 - x^*\|^2.$$  

(14)

3.2. Choosing $D$

For (6) to be efficient, the sampling vector $v$ should be sparse. For this reason we will construct $v$ so that only a (small and random) subset of its entries are non-zero.

Before we formally define $v$, let us first establish some random set terminology. Let $C \subseteq [n]$ and let $e_C := \sum_{i \in C} e_i$, where $\{e_1, \ldots, e_n\}$ are the standard basis vectors in $\mathbb{R}^n$. These subsets will be selected using a random set valued map $S$, in the literature referred to by the name *sampling* (Richtárik & Takáč 2016; Qu & Richtárik 2016).

A sampling is uniquely characterized by choosing subset probabilities $p_C \geq 0$ for all subsets $C$ of $[n]$:

$$\mathbb{P}[S = C] = p_C, \quad \forall C \subseteq [n],$$

(15)

where $\sum_{C \subseteq [n]} p_C = 1$. We will only consider proper samplings. A sampling $S$ is called proper if $p_i \overset{\text{def}}{=} \mathbb{P}[i \in S] = \sum_{C : i \in C} p_C$ is positive for all $i$.

The first analysis of a randomized optimization method with an arbitrary (proper) sampling was performed by Richtárik & Takáč (2016) in the context of randomized coordinate descent for strongly convex functions. This arbitrary sampling paradigm was later adopted in many other settings, including accelerated coordinate descent for strongly convex functions (Hanzely & Richtárik 2018), coordinate and accelerated descent for convex functions (Qu & Richtárik 2016), primal-dual methods (Qu et al. 2015; Chambolle et al. 2018), variance-reduced methods with convex (Csiba & Richtárik 2015) and nonconvex (Horváth & Richtárik 2018) objectives. Arbitrary sampling arises as a special case of our more general analysis by specializing the sampling vector to one dependent on a sampling $S$. We now define practical sampling vector $v = v(S)$ as follows:

**Lemma 3.3.** Let $S$ be a proper sampling, and let $\hat{P} = \text{Diag}(p_1, \ldots, p_n)$. Then the random vector $v = v(S)$ given by

$$v = \hat{P}^{-1} e_S$$

(16)

is a sampling vector.

**Proof.** Note that $v_i = 1_{(i \in S)}/p_i$, where $1_{(i \in S)}$ is the indicator function of the event $i \in S$. It follows that $E[v_i] = E[1_{(i \in S)}]/p_i = 1$. \hfill \Box

We can further specialize and define the following commonly used samplings. Each sampling $S$ gives rise to a particular sampling vector $v = v(S)$ (i.e., distribution $D$), which in turn gives rise to a particular stochastic reformulation (4) and SGD variant (6).

**Independent sampling.** The sampling $S$ includes every $i$, independently, with probability $p_i > 0$. This type of sampling was considered in different contexts in Horváth & Richtárik (2018); Hanzely & Richtárik (2018).

**Partition sampling.** A partition $G$ of $[n]$ is a set consisting of subsets of $[n]$ such that $\cup_{C \in G} C = [n]$ and $C_i \cap C_j = \emptyset$ for any $C_i, C_j \in G$ with $i \neq j$. A partition sampling $S$ is a sampling such that $p_C = \mathbb{P}[S = C] > 0$ for all $C \in G$ and $\sum_{C \in G} p_C = 1$.

**Single element sampling.** Only the singleton sets $\{i\}$ for $i = 1, \ldots, n$ have a non-zero probability of being sampled; that is, $\mathbb{P}[|S| = 1] = 1$. We have $\mathbb{P}[v(S) = e_i/p_i] = p_i$. 

we can calculate closed form expressions for the expected
with

By applying the above result to specific samplings, we ob-

the following smoothness assumption:

Assumption 3.4. There exists a symmetric positive defi-

for all \( x, h \in \mathbb{R}^d \) and \( i \in [n] \), where \( \| h \|^2_{M_i} := \langle M_i h, h \rangle \). In this case we say that \( f_i \) is \( M_i \)-smooth. Fur-

To better relate the above assumption to the standard smooth-

Remark 3.5. As a consequence of Assumption 3.4 we also have that each \( f_i \) is \( L_i := \lambda_{\text{max}}(M_i) \)-smooth and \( f \) is \( L := \frac{1}{n} \lambda_{\text{max}}(\sum_{i=1}^n M_i) \)-smooth. Let \( L_{\text{max}} := \max_{i \in [n]} L_i \).

Using Assumption 3.4 and a sampling we establish the fol-

Theorem 3.6. Let \( S \) be a proper sampling, and \( v = v(S) \)
(i.e., \( v \) is defined by (16)). Let \( f_i \) be \( M_i \)-smooth, and \( P \in \mathbb{R}^{n \times n} \) be defined by \( P_{i,j} := P[i \in S \land j \in S] \). Then \((f, D) \sim ES(L,\mathcal{C})\), where

For \( \tau \)-nice sampling and independent sampling, we get the

Proposition 3.8. (i) For single element sampling \( S \), we have

(ii) For independent sampling \( S \) with \( \mathbb{E}[|S|] = \tau \), we have

(iii) For \( \tau \)-nice sampling \( S \), we have

(iv) For partition sampling \( S \) with partition \( G \), we have

Generally, we do not know the values of \( h_i = \nabla f_i(x^\ast) \). But if we have prior knowledge that \( x^\ast \) belongs to some set \( \mathcal{C} \), we can obtain upper bounds for \( \sigma^2 \) for these samplings from Proposition 3.10 in a straightforward way.

4. Optimal Mini-Batch Size

Here we develop the iteration complexity for different samplings by plugging into the bounds on \( \mathcal{L} \) and \( \sigma \) given in Section 3.3 into Theorem 3.1. To keep the notation brief, in this section we drop the logarithmic term \( \log \left( \frac{2\|x^0 - x^\ast\|^2}{\epsilon} \right) \) from the iteration complexity results. Furthermore, for brevity and to better compare our results
to others in the literature, we will use \( L_i = \lambda_{\text{max}}(M_i) \) and \( L_{\text{max}} = \max_{i \in [n]} L_i \) (see Remark 3.3). Finally let \( \bar{h} = \frac{1}{n} \sum_{i \in [n]} \| h_i \|^2 \) for brevity.

**Gradient descent.** As a first sanity check, we consider the case where \( |S| = n \) with probability one. That is, each iteration (3) uses the full batch gradient. Thus \( \sigma = 0 \) and \( \tau = n \) or \( \tau = 1 \) for all \( i \) in (23) we have \( L = L_{\text{max}} \). Consequently, the resulting iteration complexity (12) is now \( \tau = n \). This is exactly the rate of gradient descent, which is precisely what we would expect since the resulting method is gradient descent.

Though an obvious sanity check, we believe this is the first convergence theorem of SGD that includes gradient descent as a special case. Clearly, this is a necessary pre-requisite if we are to hope to understand the complexity of mini-batching.

### 4.1. Nonzero gradient noise

To better appreciate how our iteration complexity evolves with increased mini-batch sizes, we now consider independent sampling with \( |S| = \tau \) and \( \tau \)-nice sampling.

**Independent sampling.** Inserting the bound on \( L \) (22) and \( \sigma \) (26) into (12) gives the following iteration complexity

\[
k \geq \frac{2}{\mu} \max \left\{ L + \max_{i \in [n]} \frac{1-p_i}{n \mu} L_i , \frac{1}{\mu} \frac{1}{n \mu} \right\}. \tag{29}
\]

This is a completely new mini-batch complexity result, which opens up the possibility of optimizing the mini-batch size and probabilities of sampling. For instance, if we fix uniform probabilities with \( p_i = \frac{\tau}{n} \) then (29) becomes

\[
l(\tau) := L + \left( \frac{\tau}{n} - \frac{1}{n} \right) L_{\text{max}}; \quad r(\tau) := \frac{2}{\mu} \left( \frac{\tau}{n} - \frac{1}{n} \right) \bar{h}. \tag{30}
\]

This complexity result corresponds to using the stepsize

\[
\gamma = \frac{1}{2} \min \left\{ \tau(\tau), \tau(\tau) \right\}. \tag{31}
\]

if \( \tau < n \), otherwise only the left-hand-side term in the minimization remains. The stepsize (31) is increasing since both \( l(\tau) \) and \( r(\tau) \) decrease as \( \tau \) increases.

With such a simple expression for the iteration complexity we can choose a mini-batch size that optimizes the total complexity. By defining the total complexity \( T(\tau) \) as the number of iterations \( k \) times the number of gradient evaluations \( (\tau) \) per iteration gives

\[
T(\tau) := \frac{2}{\mu n} \max \left\{ \tau n L + (n-\tau) L_{\text{max}}, \frac{2(n-\tau)\bar{h}}{\mu \epsilon} \right\}. \tag{32}
\]

Minimizing \( T(\tau) \) in \( \tau \) is easy because \( T(\tau) \) is a max of a linearly increasing term \( \tau \times l(\tau) \) and a linearly decreasing term \( \tau \times r(\tau) \) in \( \tau \). Furthermore \( n \times l(n) \geq 0 = n \times r(n) \). Consequently, if \( l(1) \geq r(1) \), then \( r^* = 1 \), otherwise

\[
r^* = n \frac{L_{\text{max}} - L_{\text{max}} + n L}{\mu n \bar{h}}. \tag{33}
\]

Since \( r(1) \) is proportional to the noise and \( 1/\epsilon \) and \( l(1) \) is proportional to the smoothness constants the condition \( l(1) \leq r(1) \) holds when there is comparatively a lot of noise or the precision is high. As we will see in Section 4.2, this logic extends to the case where the noise is zero, where the optimal mini-batch size is \( r^* = 1 \).

**\( \tau \)-nice sampling.** Inserting the bound on \( L \) (23) and \( \sigma \) (27) into (12) gives the iteration complexity \( k \geq \frac{2}{\mu} \max \{ l(\tau), r(\tau) \} \), where

\[
l(\tau) = \frac{n(1-\tau)}{(n-\tau)(n-1)} L + \frac{n(1-\tau)}{\tau(\tau-1)} L_{\text{max}}, \tag{34}
\]

\[
r(\tau) = \frac{2(1-\tau)}{\epsilon \mu (n-\tau)} \bar{h}, \tag{35}
\]

which holds for the stepsize

\[
\gamma = \frac{1}{2} \min \left\{ \frac{1}{l(\tau)}, \frac{1}{r(\tau)} \right\}. \tag{36}
\]

Again, this is an increasing function in \( \tau \).

We are now again able to calculate the mini-batch size that optimizes the total complexity \( T(\tau) \) given by \( T(\tau) = \frac{2}{\mu} \max \{ l(\tau), r(\tau) \} \). Once again \( T(\tau) \) is a max of a linearly increasing term \( \tau \times l(\tau) \) and a linearly decreasing term \( \tau \times r(\tau) \) in \( \tau \). Furthermore \( r(n) = 0 \leq l(n) \). Consequently, if \( r(1) \leq l(1) \) then \( r^* = 1 \), otherwise

\[
r^* = \frac{L_{\text{max}} + \frac{2}{\mu} \bar{h}}{n L - \frac{L_{\text{max}} + \frac{2}{\mu} \bar{h}}{\epsilon \mu L}}. \tag{37}
\]

### 4.2. Zero gradient noise

Consider the case where the gradient noise is zero \( \sigma = 0 \). According to Theorem 3.1 the resulting complexity of SGD with constant stepsize \( \gamma = \frac{1}{2 \epsilon} \) is given by the very simple expression

\[
k \geq \frac{2 \epsilon}{\mu}, \tag{38}
\]

where we have dropped the logarithmic term \( \log \left( \frac{\| x^0 - x^* \|^2}{\epsilon} \right) \). In this setting, due to Corollary 2.5 we know that \( f \) satisfies the weak growth condition. Thus our results are directly comparable to those developed in (Ma et al., 2018) and in (Vaswani et al., 2018).

In particular, Theorem 1 in (Ma et al., 2018) states that when running SGD with mini-batches based on sampling with replacement, the resulting iteration complexity is

\[
k \geq \frac{L_{\text{max}}}{\mu} \frac{1}{\tau} + \frac{\frac{\epsilon}{\mu} L_{\text{max}}}{\frac{1}{\tau}}, \tag{39}
\]

again dropping the logarithmic term. Now gaining insight into the complexity (38) is a matter of studying the expected smoothness parameter \( L \) for different sampling strategies.
Independent sampling. Setting $\sigma = 0$ (thus $\mathcal{T} = 0$) and using uniform probabilities with $p_i = \frac{1}{n}$ in (29) gives

$$k \geq \frac{2L}{\mu} + \left( \frac{1}{\tau} - \frac{1}{n} \right) 2L_{\max}. \quad (40)$$

$\tau-$nice sampling. If we use a uniform sampling and $\sigma = 0$ then the resulting iteration complexity is given by

$$k \geq \frac{n(\tau-1)}{\tau(n-1)} \frac{2L}{\mu} + \frac{n-\tau}{\tau(n-1)} 2L_{\max}. \quad (41)$$

Iteration complexities (39), (40) and (41) tell essentially the same story. Namely, the complexity improves as $\tau$ increases to $n$, but this improvement is not enough when considering the total complexity (multiplying by $\tau$). Indeed, for total complexity, these results all say that $\tau = 1$ is optimal.

5. Importance Sampling

In this section we propose importance sampling for single element sampling and independent sampling with $\mathbb{E}[|S|] = \tau$, respectively. Due to lack of space, the details of this section are in the appendix, Section K. Again we drop the log term in (12) and adopt the notation in Remark 3.5.

5.1. Single element sampling

For single element sampling, plugging (20) and (25) into (12) gives the following iteration complexity

$$k \geq \frac{2}{\epsilon^2 \mu} \max \left\{ \frac{\mu}{n} \max_{i \in [n]} \frac{L_i}{p_i}, \frac{2}{\epsilon^2 \mu} \sum_{i \in [n]} \frac{1}{p_i} \| h_i \|^2 \right\},$$

where $0 < p_i \leq 1$ and $\sum_{i \in [n]} p_i = 1$. In order to optimize this iteration complexity over $p_i$, we need to solve a $n$-dimensional linearly constrained nonsmooth convex minimization problem, which could be harder than the original problem (1). So instead, we will focus on minimizing $\mathcal{L}_{\max}$ and $\sigma^2$ over $p_i$, separately. We will then use these two resulting (sub)optimal probabilities to construct a sampling.

In particular, for single element sampling we can recover the partially biased sampling developed in (Needell et al., 2016). First, from (20) it is easy to see that the probabilities that minimize $\mathcal{L}_{\max}$ are $p_i^* = L_i / \sum_{j \in [n]} L_j$, for all $i$. Using these suboptimal probabilities we can construct a partially biased sampling by letting $\tilde{p}_i := \frac{1}{2} p_i^* + \frac{1}{2n}$. Plugging this sampling in (20) gives $\mathcal{L}_{\max} \leq 2 \mathcal{T} := \frac{2}{n} \max_{i \in [n]} L_i$, and from (25), we have $\sigma^2 \leq \frac{2}{n} \sum_{i \in [n]} \| h_i \|^2 := 2 \mathcal{T}$. This sampling is the same as the partially biased sampling in (Needell et al., 2016). From (29) in Theorem 3.1 we get that the total complexity is now given by

$$k \geq \max \left\{ \frac{4\mathcal{T}}{\epsilon^2 \mu}, \frac{8n}{\epsilon^2 \sigma^2} \right\}. \quad (42)$$

For uniform sampling, $\mathcal{L}_{\max} = \max_{i \in [n]} L_i \geq \mathcal{T}$ and $\sigma^2 = \frac{1}{n} \sum_{i \in [n]} \| h_i \|^2$. Hence, compared to uniform sampling, the iteration complexity of partially biased sampling is at most two times larger, but could be $n/2$ smaller in the extreme case where $L_{\max} = n \mathcal{T}$.

5.2. Minibatches

Importance sampling for minibatches was first considered in (Csiba & Richtárík, 2018), but not in the context of SGD. Here we propose the first importance sampling for mini-batch SGD. In Section K.2 in the appendix we introduce the use of partially biased sampling together with independent sampling with $|S| = \tau$ and show that we can achieve a total complexity of (by Proposition K.3)

$$k \geq \max \left\{ \left( L + \frac{2}{\tau} \mathcal{L} \right) \frac{2}{\mu}, \left( \frac{2}{\tau} - \frac{1}{n} \right) \frac{\Delta}{\epsilon^2 \mu} \right\}, \quad (43)$$

which not only eliminates the dependence on $L_{\max}$, but also improves as the mini-batch size $\tau$ increases.

6. Experiments

In this section, we empirically validate our theoretical results. We perform three experiments in each of which we highlight a different aspect of our contributions.

In the first two experiments we focus on ridge regression and regularized logistic regression problems (problems with strongly convex objective $f$ and components $f_i$) and we evaluate the performance of SGD on both synthetic and real data. In the second experiment (Section 6.2) we compare the convergence of SGD for several choices of the distribution $\mathcal{D}$ (different sampling strategies) as described in Section 3.2. In the last experiment (Section 6.3) we focus on the problem of principal component analysis (PCA) which by construction can be seen as a problem with a strongly convex objective $f$ but with non-convex functions $f_i$ (Allen-Zhu & Yuan, 2016; Garber & Hazan, 2015; Shalev-Shwartz, 2016).

In all experiments, to evaluate SGD we use the relative error measure $\frac{\| x^* - x^\dagger \|^2}{\| x^\dagger \|^2}$. For all implementations, the starting point $x^0$ is sampled from the standard Gaussian. We run each method until $\| x^k - x^\dagger \|^2 \leq 10^{-3}$ or until a pre-specified maximum number of epochs is achieved. For the horizontal axis we always use the number of epochs.

For more experiments we refer the interested reader to Section K of the Appendix.

Regularized Regression Problems: In the case of the ridge regression problem we solve:

$$\min_x f(x) = \frac{1}{2n} \sum_{i=1}^n (A[i,:;x] - y_i)^2 + \frac{1}{2} \| x \|^2,$$

while for the $L2$-regularized logistic regression problem we solve:

$$\min_x f(x) = \frac{1}{2n} \sum_{i=1}^n \log (1 + \exp(-y_i A[i,:;x])) + \frac{1}{2} \| x \|^2.$$
In both problems \( A \in \mathbb{R}^{n \times d}, y \in \mathbb{R}^n \) are the given data and \( \lambda > 0 \) is the regularization parameter. We generated synthetic data in both problems by sampling the rows of matrix \( A(A[i,:],:) \) from the standard Gaussian distribution \( \mathcal{N}(0, 1) \). Furthermore for ridge regression we sampled the entries of \( y \) from the standard Gaussian distribution while in the case of logistic regression \( y \in \{-1, 1\}^n \) where \( \mathbb{P}(y_i = 1) = \mathbb{P}(y_i = -1) = \frac{1}{2} \). For our experiments on real data we choose several LIBSVM (Chang & Lin 2011) datasets.

6.1. Constant vs decreasing step size

We now compare the performance of SGD in the constant and decreasing stepsize regimes considered in Theorems 3.1 (see (11)) and 3.2 (see (13)), respectively. Here we use a uniform single element sampling. As expected from theory, we see in Figure 1 that the decreasing stepsize regime is vastly superior at reaching a higher precision than the constant step-size variant. In our plots, the vertical red line denotes the value of \( \frac{1}{2}[L/\mu] \) predicted from Theorem 3.2 and highlights the point where SGD needs to change its update rule from constant to decreasing stepsize.

Figure 1. Comparison between constant and decreasing step size regimes of SGD. Ridge regression problem (first row): on left - synthetic data, on right - real dataset: abalone from LIBSVM. Logistic regression problem(second row): on left - synthetic data, on right - real data-set: a1a from LIBSVM. In all experiments \( \lambda = 1/n \).

Figure 2. Performance of SGD with several minibatch strategies for logistic regression. Above: the w3a data-set from LIBSVM. Below: standard Gaussian data.

In Figures 2 and 3 we compare the single element sampling (uniform and importance), \( \tau \) independent sampling (uniform, uniform with optimal batch size and importance) and \( \tau \) nice sampling (with some \( \tau \) and with optimal \( \tau^\ast \)). The probabilities of importance samplings in the single element sampling and \( \tau \) independent sampling are calculated by formulas (67) and (77) in the Appendix. Formulas for optimal minibatch size \( \tau^\ast \) in independent sampling and \( \tau \)-nice samplings are given in (53) and (57), respectively. Observe that minibatching with optimal \( \tau^\ast \) gives the best convergence. In addition, note that for constant step size, the importance sampling variants depend on the accuracy \( \epsilon \). From Figure 2 we can see that before the error reaches the required accuracy, the importance sampling variants are comparable or better than their corresponding uniform sampling variants.

6.3. Sum-of-non-convex functions

In Figure 3, our goal is to illustrate that Theorem 3.1 holds even if the functions \( f_i \) are non convex. This experiment is based on the experimental setup given in (Allen-Zhu & Yuan 2016). We first generate random vectors \( a_1, \ldots, a_n, b \in \mathbb{R}^d \) from \( U(0, 10) \) and set \( A := \frac{1}{n} \sum_{i=1}^n a_i a_i^\top \). Then we consider the problem:

\[
\min_x f(x) = \frac{1}{2n} \sum_{i=1}^n x^\top (a_i a_i^\top + D_i) x + b^\top x,
\]

where \( D_i, i \in [n] \) are diagonal matrices satisfying \( D := D_1 + \cdots + D_n = 0 \). In particular, to guarantee that \( D = 0 \), we randomly select half of the matrices and assign their \( j \)-th diagonal value \( (D_i)_{jj} \) equal to 11; for the other half we assign \( (D_i)_{jj} \) to be -11. We repeat that for all diagonal values. Note that under this construction, each \( f_i \) is a non-convex function. Once again, in the first plot we observe that while both are equally fast in the beginning, the decreasing stepsize variant is better at reaching higher accuracy than the fixed stepsize variant. In the second plot we see, as expected, that all four minibatch versions of SGD outperform single element SGD. However, while the \( \tau \)-nice and \( \tau \)-independent samplings with \( \tau = n/5 \) lead to a slight improvement only, the theoretically optimal choice \( \tau = \tau^\ast \) leads to a vast improvement.
Acknowledgements

RMG acknowledges the support by a public grant as part of the Investissement d’avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with Gaspard Monge Program for optimization, operations research and their interactions with data sciences.

References


