Optimized convergence of stochastic gradient descent by weighted averaging

Melinda Hagedorn, Heinrich Heine Univ., Düsseldorf, Germany

Florian Jarre, Heinrich Heine Univ., Düsseldorf, Germany

Sept. 23, 2022

In memory of Oleg Burdakov

Abstract

Under mild assumptions stochastic gradient methods asymptotically achieve an optimal rate of convergence if the arithmetic mean of all iterates is returned as an approximate optimal solution. However, in the absence of stochastic noise, the arithmetic mean of all iterates converges considerably slower to the optimal solution than the iterates themselves. And also in the presence of noise, when a finite termination of the stochastic gradient method is considered, the arithmetic mean is not necessarily the best possible approximation to the unknown optimal solution. This paper aims at identifying optimal strategies in a particularly simple case, the minimization of a strongly convex function with i. i. d. noise terms and finite termination. Explicit formulas for the stochastic error and the optimization error are derived in dependence of certain parameters of the SGD method. The aim was to choose parameters such that both stochastic error and optimization error are reduced compared to arithmetic averaging. This aim could not be achieved; however, by allowing a slight increase of the stochastic error it was possible to select the parameters such that a significant reduction of the optimization error could be achieved. This reduction of the optimization error has a strong effect on the approximate solution generated by the stochastic gradient method in case that only a moderate number of iterations is used or when the initial error is large. The numerical examples confirm the theoretical results and suggest that a generalization to non-quadratic objective functions may be possible.

Key words: Convex optimization, stochastic gradient descent, weighted averaging, noise, optimal step lengths, optimal weights.

1. Introduction

In Polyak and Juditsky [14] it is shown that the stochastic gradient descent algorithm asymptotically achieves optimal complexity if short but constant step lengths are chosen and if the average over all iterates is used as final output. The aim of the current paper is to explain this phenomenon with a slightly different approach and to optimize the results further by the use of weighted averages while considering finite termination and the nature of the function to be minimized.

The main source of this paper, Polyak and Juditsky [14], also inspired many other scientists to investigate weighted averages in conjunction with the stochastic gradient method in more detail. In Neu and Rosasco [13], a variant of the weighted average SGD with geometrically decreasing weights is analyzed in context of linear least-squares regression. Likewise, Cohen and Nedić [2] deal with the least-square regression. They consider constrained problems and derive upper bounds for the convergence rate and for the asymptotic ratio between convergence rate and empirical risk minimizer depending on the dimension. More abstractly, papers like Izmailov et al. [6] and Guo et al. [5], which deal with stochastic weight averaging, also build on [14].

Recently Sebbouh et al. [15] have shown almost sure convergence rates for weighted average SGD with decreasing step size. This result has already been supplemented for strongly-convex and non-convex objective functions by Liu and Yuan [10].

There certainly is also research in this area independent of Polyak and Juditsky [14]. For example, the sampling of Needell et al. [12] and the paper of Shamir and Zhang [16], in which the polynomial-decay averaging and the suffix averaging are examined.

There are numerous further modifications of the stochastic gradient approach such using momentum or heavy-ball iterations, see, e.g., [11], variance reduction [8], stochastic gradient boosting [4], and modifications tailored to specific applications. The above is merely a short and incomplete selection of related work. It seems, however, that the focus of the present paper has not been considered in this form before. This paper returns to a simple general format as considered in [14], aims at optimizing two parameters in this approach, and considers the case of an infinitely large training set in the numerical examples. A brief outline is given next.

1.1 Outline

A consequence of the results by Polyak and Juditsky [14] is that asymptotically the optimal rate of convergence of a stochastic gradient descent method is obtained when averaging *all* iterates with the same weight. This may seem counter-intuitive as one would expect the later iterates to be closer to the optimal solution and would therefore allow higher weights for later iterates. A short and intuitive explanation of why averaging over all iterates is optimal is given by the observation that the square root of the function value is reduced at a linear rate when sufficiently short steps with constant step lengths are chosen for minimizing a smooth, strongly convex function, while the reduction of the variance has a much slower

rate of convergence. Thus, asymptotically, the stochastic effects determine the overall rate of convergence, and not the condition number of the function to be minimized. And from a stochastic point of view, averaging over all iterates with the same weight is a simple but optimal strategy. However, if only a limited number of stochastic descent steps are taken, the optimization aspect and the aspect of stochastic convergence need to be balanced to each other. The present paper is an attempt to derive a simple strategy that does so in an optimized form.

To this end, an elementary derivation of the optimality result in [14] is attempted in Section 2. for a particularly simple situation, the minimization of a strongly convex quadratic function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(x) \equiv \frac{1}{m} \sum_{i=1}^{m} f_i(x) \equiv \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} x^{\mathrm{T}} A^{(i)} x + (b^{(i)})^{\mathrm{T}} x + c_i \tag{1}$$

by a stochastic gradient approach. Exact and computable formulas for the variance of the averaged iterates as a measure of the *stochastic error* and for the contraction constant of the descent steps as a measure of the *optimization error* of the weighted iterates are derived. Given these explicit formulas standard nonlinear minimization algorithms with different starting points were applied to reduce both errors simultaneously by adjusting certain parameters associated with the weights and the step lengths of a stochastic gradient method. While it turned out that the goal could not be reached of reducing both, optimization error and stochastic error at the same time, a significant reduction of the optimization error was possible by allowing a slight increase (e.g. of 10%) of the stochastic error.

Generalizations to further classes of smooth convex functions are discussed in Section 3..

1.2 Notation

The condition number with respect to the 2-norm of a square matrix A is denoted by $\mathrm{cond}(A)$ and the smallest and largest eigenvalues are denoted by $\lambda_{min}(A)$ and $\lambda_{max}(A)$. The Hadamard product (componentwise product) of two matrices X, Y with the same dimensions is denoted by $X \circ Y$ and powers of vectors are also defined componentwise, e.g. $x^2 = x \circ x$. The diagonal of a square matrix A is denoted by $\mathrm{diag}(A)$ and a diagonal matrix with diagonal $x \in \mathbb{R}^n$ is denoted by $\mathrm{Diag}(x)$. The i-th canonical unit vector is denoted by e_i and the all-one-vector is denoted by e, its dimension being given by the context. Finally, let $\mathbb{1}_{n \times n}$ denote the $n \times n$ matrix with all entries equal to one.

1.3 A Standard Stochastic Descent Method

For large values of m (or when $m = \infty$) a stochastic gradient descent method of the following form is considered: Assume that a batch S_k is chosen i. i. d. from the uniform distribution of $\{1, \ldots, m\}$. Then, as is well known, the expected value of the gradient of $f_{S_k}(x^k) :=$

$\frac{1}{|S_k|} \sum_{i \in S_k} f_i(x^k)$ is the full gradient,

$$E(\nabla f_{S_k}(x^k)) = E\left(\frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(x^k)\right) = \nabla f(x^k).$$

This motivates the stochastic gradient descent that uses the approximation $\nabla f_{S_k}(x^k)$ instead of $\nabla f(x^k)$ to define a sequence of iterates. Let $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ be a weakly monotonously decreasing function (used in the analysis below) and consider a-priori-defined[1] step lengths $\gamma_k = \gamma(k)$ in a stochastic gradient descent approach

$$x^{k+1} = x^k - \gamma_k \nabla f_{S_k}(x^k) = x^k - \gamma_k(\nabla f(x^k) + \boldsymbol{\xi}^k) \tag{2}$$

with random noise term

$$\boldsymbol{\xi}^k := \nabla f_{S_k}(x^k) - \nabla f(x^k) = \frac{1}{|S_k|} \sum_{i \in S_k} A^{(i)} x^k + b^{(i)} - \frac{1}{m} \sum_{i=1}^m A^{(i)} x^k + b^{(i)} \tag{3}$$

whose expected value is zero.

1.3.1 Assumptions

For simplicity it is assumed that the batch size $|S_k|$ is constant for all k. As $\boldsymbol{\xi}^k$ depends on the current iterate, and thus on previous noise terms $\boldsymbol{\xi}^i$ for $i < k$ it may seem unrealistic to assume that the terms $\boldsymbol{\xi}^k$ are i.i.d. as well. However, setting

$$\Delta A_{S_k} := \frac{1}{|S_k|} \sum_{i \in S_k} A^{(i)} - \frac{1}{m} \sum_{i=1}^m A^{(i)} \quad \text{and} \quad \Delta b_{S_k} := \frac{1}{|S_k|} \sum_{i \in S_k} b^{(i)} - \frac{1}{m} \sum_{i=1}^m b^{(i)}$$

the noise can be written as
$$\boldsymbol{\xi}^k = \Delta A_{S_k} x^k + \Delta b_{S_k}$$
where both ΔA_{S_k} and Δb_{S_k} are i.i.d. and the expected values satisfy

$$E(\Delta A_{S_k}) = 0 \in \mathbb{R}^{n \times n} \quad \text{and} \quad E(\Delta b_{S_k}) = 0 \in \mathbb{R}^n.$$

Since $|S_k|$ is constant for all k, also $\sigma_A^2 := E(\|\Delta A_{S_k}\|_F^2)$ and $\sigma_b^2 := E(\|\Delta b_{S_k}\|_2^2)$ are independent of k. Moreover, when the iterates x^k converge to some limit x^* then, asymptotically, the error terms

$$\boldsymbol{\xi}^k \quad \text{are i.i.d.} \tag{4}$$

This is the situation that is also analyzed in a different context as part (c) of Theorem 1 in [14], and this assumption will be used for simplicity below.

---

[1]In the presence of noise, a line search is difficult.

For the analysis it can be assumed further, without loss of generality, that $x^* = 0$ and that the Hessian of f is a diagonal matrix D with diagonal elements $0 < D_{1,1} \leq \ldots \leq D_{n,n}$. Thus, $\nabla f(x) = Dx$.

Finally, assume that some upper bound for the largest eigenvalue $D_{n,n}$ is known so that the step lengths γ_k can be chosen in the half-open interval $(0, 1/D_{n,n}]$. (A positive lower bound for $D_{1,1}$ is not assumed to be known.)

Up to a factor of 2, the bound on γ_k is essentially the bound in [14]. (Note that there is a minor error in [14]: in Assumption 2.2 (page 839) they write $2/\min_i \mathrm{Re}(D_{i,i})$ while the correct statement would be $\min_i 2\mathrm{Re}(D_{i,i})/|D_{i,i}|^2$ in order for the argument at the bottom of page 844 of [14] to be valid.)

1.4 A Simple Algorithm with Two Parameters

In Polayk and Juditsky [14], an algorithm is considered with short step lengths and with output given by the arithmetic mean of all iterates x^k generated via (2). In the absence of noise, i.e. in case that all terms $\boldsymbol{\xi}^k$ are zero, it is clear that the final iterate x^k is a much better approximation to the optimal solution than the average. On the other hand, as shown in [14], averaging all iterates with equal weights reduces the variance, and is asymptotically optimal for large k. In the following a method is considered that uses weighted averages with higher weights on later iterates. When the noise is small, the weighted average also is a better approximation to the optimal solution than the average with equal weights. To reduce the variance of the iterates with higher weight, a possible step length reduction for the later iterates is considered.

In the following it is assumed that an initial iterate x^0 is given and that iterates x^k are generated via (2). It is further assumed that the output of the algorithm is given by weighted averages $\bar{x}^k$ defined as

$$\bar{x}^k := \left(\sum_{j=1}^{k} w_j \right)^{-1} \sum_{j=1}^{k} w_j x^j \tag{5}$$

and that $\gamma_k \equiv \gamma(k)$ in (2), and $w_j \equiv w(j)$ where

$$\begin{aligned} \gamma(t) &\equiv c \left(\tfrac{M}{t+M} \right)^{\alpha}, \\ w(t) &\equiv t^{\beta}, \end{aligned} \quad \text{for some} \quad \alpha \geq 0, \ \beta \geq 0, \ 0 < c \leq \frac{1}{D_{n,n}}, \ M \geq 1. \tag{6}$$

The asymptotic analysis in [14] (Assumption 2.2) covers the case $\alpha \in (0,1)$, $\beta = 0$ and $M = 1$. Larger values of α lead to a faster reduction of the step length, and for $M > 1$ the step length reduction in the early iterations is slower. In the numerical experiments the choice $M = 1 + \delta k^{max}$ is considered where k^{max} is the value of k at which the iteration (5) is stopped, and $\delta \in [0,1]$ is a parameter to be determined. If $\delta > 0$ is fixed, the step lengths γ do not converge to zero when $k^{max} \to \infty$. It is the aim of this paper to determine optimal values α, c, M of the step lengths and an optimal value β of the weights for finite values of k^{max}.

For completeness, the algorithm outlined above is stated in detail:

Algorithm 1
Input: $x^0 \in \mathbb{R}^n$, $\gamma(.)$, $w(.)$ as in (6), constant batch size ν, number of iterations k^{max}.
Initialization: Set $\sigma := 0$ (sum of the weights) and $\bar{x}^0 := 0$ (weighted sum of iterates).

For $k = 0, 1, \ldots, k^{max} - 1$ do

  1. Select a batch S_k of size $|S_k| = \nu$ i.i.d. from $\{1, \ldots, m\}$.

  2. Set $x^{k+1} := x^k - \gamma(k)\nabla f_{S_k}(x^k)$.

  3. Set $\bar{x}^{k+1} := \bar{x}^k + w(k+1)x^{k+1}$ and $\sigma := \sigma + w(k+1)$.

Set the weighted average $\bar{x}^{final} := \bar{x}^{k^{max}}/\sigma$ and return $\bar{x}^{final}$ as approximate minimizer of f.

For $\alpha = \beta = 0$, Algorithm 1 reduces to an algorithm with asymptotically optimal parameters in [14]. In order to identify optimal parameters depending on the number of iterations allocated in advance an elementary analysis of the iterates is presented next.

2. Optimal Selection of Parameters

2.1 Analysis Without Averaging

With the above assumptions,

$$x^1 = x^0 - \gamma_0(\nabla f(x^0) + \boldsymbol{\xi}^0) = x^0 - \gamma_0 D x^0 - \gamma_0 \boldsymbol{\xi}^0 = (I - \gamma_0 D)x^0 - \gamma_0 \boldsymbol{\xi}^0$$

where $(I - \gamma_0 D)$ is a contraction satisfying

$$0 \preceq (1 - \gamma_0 D_{n,n})I \preceq (I - \gamma_0 D) \preceq (1 - \gamma_0 D_{1,1})I \prec I.$$

For the next step, there is new noise "$-\gamma_1 \boldsymbol{\xi}^1$" while the noise of the previous iteration is reduced,

$$x^2 = (I - \gamma_1 D)x^1 - \gamma_1 \boldsymbol{\xi}^1 = (I - \gamma_1 D)[(I - \gamma_0 D)x^0 - \gamma_0 \boldsymbol{\xi}^0] - \gamma_1 \boldsymbol{\xi}^1$$
$$= (I - \gamma_1 D)(I - \gamma_0 D)x^0 - (I - \gamma_1 D)\gamma_0 \boldsymbol{\xi}^0 - \gamma_1 \boldsymbol{\xi}^1.$$

Denote the contraction $I - \gamma_i D$ by C_i and observe that $\gamma_i \in (0, 1/D_{n,n}]$ implies $||C_i|| = 1 - \gamma_i D_{1,1} < 1$.

Let the product of contractions be denoted by

$$C_{i,k} = \prod_{j=i}^{k} C_j = \prod_{j=i}^{k}(I - \gamma_i D), \tag{7}$$

6

where the empty product $C_{i,k}$ for $i > k$ is equal to I by convention, for example, $C_{k+1,k} = I$. Since the largest entry of all C_i is at the first diagonal position, it follows that $\|C_{i,k}\| = \prod_{j=i}^{k}(1 - \gamma_j D_{1,1})$. Again, the empty product is 1, i.e. $\|C_{k+1,k}\| = 1$. The process (2) then leads to the aggregated representation

$$x^k = \left[\prod_{i=0}^{k-1}(I - \gamma_i D)\right] x^0 - \sum_{i=0}^{k-1} \left[\prod_{j=i+1}^{k-1}(I - \gamma_j D)\right] \gamma_i \boldsymbol{\xi}^i = C_{0,k-1} x^0 - \sum_{i=0}^{k-1} \gamma_i C_{i+1,k-1} \boldsymbol{\xi}^i \quad (8)$$

with the expected value of x^k given by $E(x^k) = C_{0,k-1} x^0$ and noise term $\sum_{i=0}^{k-1} \gamma_i C_{i+1,k-1} \boldsymbol{\xi}^i$. Observe that $E(x^k)$ coincides with the "ideal" iterates, meaning the iterates without noise and denote the "ideal" iterates by

$$\breve{x}^k := C_{0,k-1} x^0 \quad \text{for} \quad k \geq 1. \quad (9)$$

2.2 Analysis With Averaging

Now, consider weighted averaged iterates (5) for the weight function $w : \mathbb{R}_+ \to \mathbb{R}_+$ given in (6) and $w_j = w(j)$. Denote the "ideal" averages (without noise) by

$$\breve{\bar{x}}^k := \left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{j=1}^{k} w_j \breve{x}^j = \left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{j=1}^{k} w_j C_{0,j-1} x^0 \quad (10)$$

and the "accumulated" noise in x^k by

$$\breve{\boldsymbol{\xi}}^k := x^k - \breve{x}^k.$$

The noise satisfies the recurrence relation $\breve{\boldsymbol{\xi}}^0 := 0$ and

$$\begin{aligned}
\breve{\boldsymbol{\xi}}^{k+1} &= x^{k+1} - \breve{x}^{k+1} \overset{(9)}{=} x^{k+1} - C_{0,k} x^0 \\
&\overset{(8)}{=} C_{0,k} x^0 - \sum_{i=0}^{k} \gamma_i C_{i+1,k} \boldsymbol{\xi}^i - C_{0,k} x^0 = -\sum_{i=0}^{k} \gamma_i C_{i+1,k} \boldsymbol{\xi}^i. \quad (11)
\end{aligned}$$

For indices $i \in \{0, \ldots, k-1\}$ and $j \in \{1, \ldots, k\}$ with $k \in \mathbb{N}$ a reordering of the sum

$$\sum_{j=1}^{k} \sum_{i=0}^{j-1} a_{i,j} = \sum_{i=0}^{k-1} \sum_{j=i+1}^{k} a_{i,j}. \quad (12)$$

leads to a representation of the "total" noise $\bar{x}^k - \breve{x}^k$ in $\bar{x}^k$ denoted by $\overset{\smile}{\boldsymbol{\xi}}^k$,

$$
\begin{aligned}
\overset{\smile}{\boldsymbol{\xi}}^k := \bar{x}^k - \breve{x}^k \ &= \ \left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{j=1}^{k} w_j (x^j - \breve{x}^j) \\
&\overset{(11)}{=} \ -\left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{j=1}^{k} w_j \sum_{i=0}^{j-1} \gamma_i C_{i+1,j-1}\,\boldsymbol{\xi}^i \\
&\overset{(12)}{=} \ -\left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{i=0}^{k-1} \sum_{j=i+1}^{k} w_j \gamma_i C_{i+1,j-1}\,\boldsymbol{\xi}^i \\
&= \ \left(\sum_{j=1}^{k} w_j\right)^{-1} \sum_{i=0}^{k-1} G_{i,k}\,\boldsymbol{\xi}^i
\end{aligned}
$$

where

$$
G_{i,k} := -\gamma_i \sum_{j=i+1}^{k} w_j C_{i+1,j-1} \qquad \text{for} \qquad 0 \le i \le k-1. \tag{13}
$$

2.3 Parameter Selection Without Averaging

For a comparison, first an algorithm is considered that generates the "plain" iterate $x^{k^{max}}$ as output rather than the weighted average $\bar{x}^{final}$.

In this case, based on (8), the quantity to be minimized in the design of a method with optimal convergence would be a weighted sum of $\|C_{0,k-1}\|$ and of a bound of the variance of the noise term. Due to stochastic independence of the noise terms (4), the covariance matrix of x^k satisfies

$$
\mathrm{Cov}\left(\sum_{i=0}^{k-1} \gamma_i C_{i+1,k-1} \boldsymbol{\xi}^i\right) = \sum_{i=0}^{k-1} \gamma_i^2 \left(C_{i+1,k-1}\right) \mathrm{Cov}(\boldsymbol{\xi}^i) \left(C_{i+1,k-1}\right)^{\mathrm{T}}.
$$

Furthermore, by (4) the covariance matrices $\mathrm{Cov}(\boldsymbol{\xi}^i)$ are all equal, $\mathrm{Cov}(\boldsymbol{\xi}^i) \equiv \Sigma$ for some positive semidefinite Σ. In the case that Σ is a multiple of the identity matrix, one obtains an exact reformulation

$$
\left\|\mathrm{Cov}\left(\sum_{i=0}^{k-1} \gamma_i C_{i+1,k-1} \boldsymbol{\xi}^i\right)\right\| = \left\|\sum_{i=0}^{k-1} \gamma_i^2 \left(C_{i+1,k-1}\right) \Sigma \left(C_{i+1,k-1}\right)\right\| = \sum_{i=0}^{k-1} \gamma_i^2 \|C_{i+1,k-1}\|^2 \|\Sigma\|
$$

since all $C_{i+1,k-1}$ are diagonal matrices with their largest entry defining their norm at the $(1,1)$-position, and in the case of a general positive semidefinite matrix Σ one obtains an upper bound

$$
\left\|\mathrm{Cov}\left(\sum_{i=0}^{k-1} \gamma_i C_{i+1,k-1} \boldsymbol{\xi}^i\right)\right\| = \left\|\sum_{i=0}^{k-1} \gamma_i^2 \left(C_{i+1,k-1}\right) \Sigma \left(C_{i+1,k-1}\right)\right\| \le \sum_{i=0}^{k-1} \gamma_i^2 \|C_{i+1,k-1}\|^2 \|\Sigma\| \tag{14}
$$

where the inequality follows from a repeated application of the triangle inequality and from sub-multiplicativity of the norm. Let

$$\vartheta(\gamma) := \left(\sum_{i=0}^{k-1} \gamma_i^2 \, \|C_{i+1,k-1}\|^2 \right)^{1/2}.$$ (15)

Optimizing the expected norm of x^k

$$E\left(\left\|x^k\right\|\right) \le \|C_{0,k-1}\| \|x^0\| + \vartheta(\gamma) \, \|\Sigma\|^{1/2}$$ (16)

thus leads to the aim of choosing the step length function $\gamma(\,.\,)$ such that

$$\left(\, \|C_{0,k-1}\| + \mu \, \vartheta(\gamma) \, \right) / (1+\mu)$$ (17)

is minimized where $\mu \ge 0$ is a fixed weight. For $\mu := \|\Sigma\|^{1/2} / \|x^0\|$ the minimizers of (16) and (17) coincide; but unfortunately, the ratio of "noise" $\|\Sigma\|^{1/2}$ to "starting error" $\|x^0\|$ generally is not known. Nevertheless, the separation of "optimization error" (here $\|C_{0,k-1}\| \|x^0\|$) and "stochastic error" (here $\vartheta(\gamma) \, \|\Sigma\|^{1/2}$) can be extended to weighted averages in the next sub-section and will then be exploited with the aim of identifying suitable parameters α, β, δ, M.

2.4 Parameter Selection With Averaging

Since the step lengths and weights are pre-defined the same is true for $G_{i,k}$, and thus, since the noise terms $\boldsymbol{\xi}^i$ are assumed to be i.i.d., also $G_{i,k}\boldsymbol{\xi}^i$ are independently distributed. The variance of $\breve{\boldsymbol{\xi}}^k$ therefore is $\left(\sum_{j=1}^{k} w_j \right)^{-2}$ times the sum of the variances of $G_{i,k}\,\boldsymbol{\xi}^i$. The latter are bounded by a fixed multiple (depending on the distribution of the terms $\boldsymbol{\xi}^i$) of $\|G_{i,k}\|^2$.

To reduce the expected value of $\|\breve{\boldsymbol{\xi}}^k\|_2$, it is therefore the aim to define the weight function w and the step length function γ such that

$$\kappa(w,\gamma) := \left(\left(\sum_{j=1}^{k} w_j \right)^{-2} \sum_{i=0}^{k-1} \|G_{i,k}\|^2 \right)^{1/2}$$ (18)

is small. As in the case of ϑ in (15), the upper bound κ^2 is the exact norm of the covariance matrix of $\breve{\boldsymbol{\xi}}^k$ when Σ is a multiple of the identity matrix.

Simultaneously, the second goal is that also the norm of $\breve{x}^k$ should be small, i.e.

$$\tau(w,\gamma) := \left\| \left(\sum_{j=1}^{k} w_j \right)^{-1} \sum_{j=1}^{k} w_j \prod_{i=0}^{j-1}(I - \gamma_i D) \right\| = \left(\sum_{j=1}^{k} w_j \right)^{-1} \sum_{j=1}^{k} w_j \, \|C_{0,j-1}\|$$ (19)

9

should be small. The above equation again uses the fact that all C_i are diagonal matrices with the largest entry defining the norm at the $(1,1)$ position. Thus, in place of (17) it is the aim of choosing the step length function $\gamma(.)$ and the weight function $w(.)$ such that

$$r(w,\gamma) := (\tau(w,\gamma) + \mu\kappa(w,\gamma))/(1+\mu) \tag{20}$$

is minimized for a given fixed $\mu \geq 0$. Again, an "appropriate" choice of μ is not evident.

For the special case (6) the evaluation of κ and τ with order k arithmetic operations is considered next.

2.4.1 Evaluation of All Norms $\|C_{0,j}\|$ and $\|C_{i+1,k-1}\|$

The choice of $\gamma \in (0, 1/D_{n,n}]$ implies that

$$\|C_k\|_2 = \|I - \gamma_k D\|_2 = 1 - \gamma_k D_{1,1} = 1 - \bar{c}\left(\frac{M}{k+M}\right)^\alpha \qquad \text{for all } k \geq 0 \tag{21}$$

where

$$\bar{c} := cD_{1,1} \leq \frac{D_{1,1}}{D_{n,n}} = \frac{1}{\text{cond}(D)}.$$

For $0 \leq j \leq k$ note that

$$\|C_{0,j}\| = \|\prod_{\ell=0}^{j} C_\ell\| = \prod_{\ell=0}^{j}\left(1 - \bar{c}\left(\frac{M}{\ell+M}\right)^\alpha\right) = \|C_{0,j-1}\|\left(1 - \bar{c}\left(\frac{M}{j+M}\right)^\alpha\right)$$

where the second equality again follows from the diagonal structure of $C_\ell = I - \gamma_\ell D$ with the largest entry in absolute value always at the $(1,1)$-position.

Thus, starting from $\|C_{0,0}\| = 1 - \bar{c}$, all $\|C_{0,j}\|$ can be computed for $j = 1, 2, 3 \ldots, k$ with a total of order k arithmetic operations.

Likewise, starting with $\|C_{k,k-1}\|$, which is 1 by convention, the predecessors $\|C_{j,k-1}\|$ can be iteratively determined by

$$\|C_{j,k-1}\| = \prod_{\ell=j}^{k-1}\left(1 - \bar{c}\left(\frac{M}{\ell+M}\right)^\alpha\right) = \|C_{j+1,k-1}\|\left(1 - \bar{c}\left(\frac{M}{j+M}\right)^\alpha\right)$$

for $j \in \{k-1, \ldots, 0\}$, and ϑ in (15) can be evaluated with order k arithmetic operations.

2.4.2 Evaluation of All Norms $\|G_{i,k}\|$

The aim of this section is to derive a scheme for evaluating all norms $\|G_{i,k}\|$ for $0 \leq i \leq k-1$ also with order k arithmetic operations, so that the parameters α, β, c, and M can easily be optimized for maximum iteration numbers k^{max} up to the order of about 10^8.

Note that for $0 \leq i \leq k-1$:

$$\|G_{i,k}\| = \left\| \gamma_i \sum_{j=i+1}^{k} w_j C_{i+1,j-1} \right\| = \gamma_i \left\| \sum_{j=i+1}^{k} j^\beta C_{i+1,j-1} \right\| = c \left(\frac{M}{i+M} \right)^\alpha \sum_{j=i+1}^{k} j^\beta \|C_{i+1,j-1}\| \quad (22)$$

where the last equality follows again since all C_i are diagonal matrices with the largest entry defining the norm at the (1,1) position. Since $\|C_{i+1,i}\| = 1$ it follows that

$$\sum_{j=i+1}^{k} j^\beta \|C_{i+1,j-1}\| = (i+1)^\beta + \sum_{j=i+2}^{k} j^\beta \|C_{i+1,j-1}\| = (i+1)^\beta + \sum_{j=i+2}^{k} j^\beta \|C_{i+1}\| \|C_{i+2,j-1}\|$$

$$= (i+1)^\beta + \|C_{i+1}\| \sum_{j=i+2}^{k} j^\beta \|C_{i+2,j-1}\| = (i+1)^\beta + \|C_{i+1}\| \left(\frac{(i+1+M)^\alpha}{cM^\alpha} \|G_{i+1,k}\| \right).$$

In the last equation relation (22) has been used for $\|G_{i+1,k}\|$ in place of $\|G_{i,k}\|$. Hence, starting with $\|G_{k-1,k}\| = \frac{ck^\beta M^\alpha}{(k-1+M)^\alpha}$, all $\|G_{i,k}\|$ can be computed for $i = k-2, k-3, \ldots$ via

$$\|G_{i,k}\| = c \left(\frac{M}{i+M} \right)^\alpha (i+1)^\beta + \|C_{i+1}\| \left(\frac{i+1+M}{i+M} \right)^\alpha \|G_{i+1,k}\|$$

with $\|C_{i+1}\| = 1 - \frac{\bar{c}M^\alpha}{(i+1+M)^\alpha}$.

For the case $\alpha = \beta = 0$ as considered in [14], the above simplifies to $\|G_{k-j,k}\| = c(1 - (1-\bar{c})^j)/\bar{c}$, and the quantities κ and τ allow a closed form representation,

$$\kappa = \frac{c}{k^{max}\,\bar{c}} \left(k^{max} - 2\frac{1 - \bar{c} - (1-\bar{c})^{k^{max}+1}}{\bar{c}} + \frac{(1-\bar{c})^2 - (1-\bar{c})^{2k^{max}+2}}{1 - (1-\bar{c})^2} \right)^{1/2}$$

and

$$\tau = \frac{(1-\bar{c})(1 - (1-\bar{c})^{k^{max}})}{k^{max}\,\bar{c}}.$$

The straightforward derivation of these formulas is omitted for brevity. For fixed values of $\bar{c} > 0$ it follows that κ is of the order $1/\sqrt{k^{max}}$ and τ is of the order $1/k^{max}$ so that convergence of Algorithm 1 to the optimal solution when $k^{max} \to \infty$ follows also for the case $\alpha = \beta = 0$.

With these preparations it is now possible to evaluate $\tau(w, \gamma)$ and $\kappa(w, \gamma)$ in (19) and (18) with order k arithmetic operations and thus to minimize the function r with w, γ as in (6). Here, larger values of μ are meaningful, when the noise is large compared to the distance of the initial iterate from optimality.

Note that when f is multiplied by some constant $\eta > 0$ then $\bar{c}$ and all $\|C_{i,j}\|$ remain invariant, but c and all $\|G_{i,k}\|$ are multiplied by $1/\eta$. Hence, when multiplying μ with η, the minimizer of (20) remains invariant.

11

3. Numerical Examples

3.1 Optimizing the Parameters of Weighted Averaging

In this sub-section the selection of the parameters α, β, c, and M in algorithm 1 is considered minimizing the function r in (20). Here, r is a weighted sum of τ and κ where $\tau \geq 0$ always is less than 1 and decreases with increasing values of β. On the other hand, κ is minimized for $\beta = 0$. It is the aim of the considerations below to balance the conflicting goals of minimizing both τ and κ.

To standardize the results in this sub-section the largest eigenvalue of f is fixed to

$$D_{n,n} = 1$$

throughout.

Table 1 and Table 2 give some intuition about the values of τ and κ when $\alpha = \beta = 0$ and c is also fixed to $c = 1$.

k^{max} \ $cond(D)$	$10^{0.5}$	10^1	$10^{1.5}$	10^2	$10^{2.5}$	10^3	$10^{3.5}$	10^4
10^2	-1.6646	-1.0458	-0.5315	-0.2023	-0.0676	-0.0218	-0.0069	-0.0022
$10^{2.5}$	-2.1643	-1.5454	-1.0133	-0.5226	-0.1999	-0.0671	-0.0216	-0.0069
10^3	-2.6646	-2.0458	-1.5136	-1.0044	-0.5198	-0.1995	-0.0669	-0.0216
$10^{3.5}$	-3.1643	-2.5454	-2.0133	-1.5041	-1.0008	-0.5189	-0.1990	-0.0668
10^4	-3.6646	-3.0458	-2.5136	-2.0044	-1.5011	-1.0005	-0.5186	-0.1992
$10^{4.5}$	-4.1643	-3.5454	-3.0133	-2.5041	-2.0007	-1.5001	-0.9995	-0.5186
10^5	-4.6646	-4.0458	-3.5136	-3.0044	-2.5011	-2.0004	-1.4998	-1.0001
$10^{5.5}$	-5.1643	-4.5454	-4.0133	-3.5041	-3.0007	-2.5001	-1.9995	-1.4997
10^6	-5.6646	-5.0458	-4.5136	-4.0044	-3.5011	-3.0004	-2.4998	-2.0000
$10^{6.5}$	-6.1643	-5.5454	-5.0133	-4.5041	-4.0007	-3.5001	-2.9995	-2.4997
10^7	-6.6646	-6.0458	-5.5136	-5.0044	-4.5011	-4.0004	-3.4998	-3.0000
$10^{7.5}$	-7.1643	-6.5454	-6.0133	-5.5041	-5.0007	-4.5001	-3.9995	-3.4997
10^8	-7.6646	-7.0458	-6.5136	-6.0044	-5.5011	-5.0004	-4.4998	-4.0000

Table 1: Values of $\log_{10}(\tau(0,0))$ for $k^{max} = 10^2$, $10^{2.5}$, 10^3, ..., 10^8 in rows $1-13$ and condition numbers $10^{0.5}$, 10^1, $10^{1.5}$, ... 10^4 in columns $1-7$

The values of $\tau(0,0)$ in Table 1 are not surprising, and are listed only as a comparison to Table 2 below. When k^{max} is larger than the condition number then $\tau(0,0)$ is roughly of the order $\frac{\text{condition number}}{k^{max}}$.

In Table 2 below it is interesting to observe that for large condition numbers such as condition number 10^4 in column 7, the bound $\kappa(0,0)$ for the variance increases first when k^{max} increases, and starting from $k^{max} = 100$ it reaches a maximum of $\kappa(0,0) \approx 41.7920$ for $k^{max} = 10^{4.5}$ in row 6 before decreasing again for larger values of k^{max}. (For the dimension, and below also for k^{max}, square roots are rounded to integer values in Table 1 and Table 2.)

$cond(D)$ / k^{max}	$10^{0.5}$	10^1	$10^{1.5}$	10^2	$10^{2.5}$	10^3	$10^{3.5}$	10^4
10^2	0.3109	0.9288	2.3730	4.1385	5.1887	5.6063	5.7490	5.7952
$10^{2.5}$	0.1770	0.5502	1.6449	4.1915	7.3131	9.1714	9.9138	10.1670
10^3	0.0999	0.3140	0.9773	2.9176	7.4412	12.9772	16.2857	17.6052
$10^{3.5}$	0.0563	0.1775	0.5588	1.7365	5.1907	13.2193	23.0643	28.9336
10^4	0.0316	0.0999	0.3157	0.9925	3.0887	9.2203	23.5113	41.0028
$10^{4.5}$	0.0178	0.0562	0.1779	0.5612	1.7668	5.4904	16.4110	41.7920
10^5	0.0100	0.0316	0.1000	0.3160	0.9983	3.1385	9.7669	29.1551
$10^{5.5}$	0.0056	0.0178	0.0563	0.1779	0.5625	1.7747	5.5871	17.3619
10^6	0.0032	0.0100	0.0316	0.1000	0.3164	0.9993	3.1570	9.9247
$10^{6.5}$	0.0018	0.0056	0.0178	0.0563	0.1780	0.5624	1.7789	5.6121
10^7	0.0010	0.0032	0.0100	0.0316	0.1001	0.3162	1.0005	3.1599
$10^{7.5}$	0.0006	0.0018	0.0056	0.0178	0.0563	0.1779	0.5629	1.7785
10^8	0.0003	0.0010	0.0032	0.0100	0.0316	0.1000	0.3164	0.9999

Table 2: Values of $\kappa(0,0)$ for $k^{max} = 10^2,\ 10^{2.5},\ 10^3,\ \ldots,\ 10^8$ in rows $1-13$
and condition numbers $10^{0.5},\ 10^1,\ 10^{1.5},\ \ldots 10^4$ in columns $1-7$

While condition numbers as large as 10^4 might not be typical for stochastic applications, and while moderate iteration numbers such as $k^{max} = 10^{4.5}$ cannot render significant progress for such large condition numbers, it is interesting to consider possible improvements of κ for $k^{max} = 10^{4.5}$ and $D_{1,1} = 10^{-4}$ by optimizing $\tau + \mu\,\kappa$ with respect to α, β, c, and M for different values of μ.

For minimizing the function $r = (\tau + \mu\,\kappa)/(1+\mu)$ in (20) the descent algorithm "min_f.m" from [9] was used that aims for a local minimizer near the starting point:

The variables α and β were constrained to the intervals $[0,\ 2]$ and $[0,\ 5]$. (The upper bounds $\alpha \leq 2$ and $\beta \leq 5$ were chosen at will to limit the search space to a compact domain.) For $c \leq 1$ a lower bound of 0.1 was chosen. (The reduction of τ is considered as too slow when $c < 0.1$.) Finally M was set as $1 + \delta k^{max}$ with $\delta \in [0,1]$. In Table 3, the (approximately) optimal parameters identified with min_f are listed for different weights $\mu > 0$ and $k^{max} = 10^{4.5}$, $D_{1,1} = 10^{-4}$.

For each run of min_f, the four starting values $(\alpha, \beta) = (0,0),\ (0,2),\ (\frac{1}{2},0),\ (\frac{1}{2},2)$ were used as well as $\delta = 0.1$ and $c = 0.5$. When min_f identified different approximate optimal solutions the one with the lowest value of $r(\alpha, \beta)$ is listed.

μ	NA	1	0.1	0.017	0.017	0.01	0.001
α	0	2	2	2	0	0	0
β	0	0	0	0	0.718	2.081	5
c	1	0.1	0.1	0.1	1	1	1
δ	0	0	0	0	0	0	0
τ	0.303	1.000	1.000	1.000	0.189	0.114	0.073
κ	41.79	0.104	0.104	0.104	47.60	53.18	58.84
r		0.552	0.919	0.985	0.982	0.639	0.132

Table 3: Optimal parameters α, β, c, δ for $k^{max} = 10^{4.5}$, $D_{1,1} = 10^{-4}$, and different values of μ.

The numbers in the first column refer to the situation of Polyak and Juditsky [14] and are not the result of an optimization process. The two columns for $\mu = 0.017$ show the situation that two approximate local minimizers were found with similar objective value "r" but rather different input arguments. The optimal parameters appear to be discontinuous near $\mu = 0.017$. Since the "appropriate" weight $\mu > 0$ depends on the unknown distance of x^0 to the optimal solution and on the unknown magnitude of the noise, the selection of α, β, c, and δ cannot be extracted from data such as Table 3, even if $D_{1,1}$ and $D_{n,n}$ are known.

Given an example that is not as poorly conditioned as in Table 3, namely

$$D_{1,1} = 0.03,$$

different values of k^{max} and weighting terms μ were considered for Table 4 such that at least two approximate local optimal solutions α, β, c, δ could be identified.

k^{max}	1000		10000		100000		
μ	0.05		0.012		0.00148		
α	1.104	0	0.519	0	1.195	0	0
β	1.382	0.809	0.614	0.606	5	0.5955	0.6521
c	1	1	1	1	1	1	1
δ	0.186	0	0.164	0	1.29e-3	0	0
τ	2.37e-3	3.49e-3	0.3594	0.3593	5.92e-7	3.94e-6	2.61e-6
κ	1.171	1.152	1.45e-4	1.47e-4	0.119	0.114	0.115
r	5.80e-2	5.82e-2	4.4054e-3	4.4057e-3	1.66e-4	1.72e-4	1.72e-4

Table 4: Nearly optimal parameters α, β, c, δ for different values of μ and k^{max}.

Because of the discontinuous dependence of α and β (and also of τ and κ) on μ as observed in Table 3 and Table 4, another selection process for optimizing the parameters α, β, c, δ was considered, namely

$$\min_{-1 \leq v_1, v_2 \leq 0.1} v_1 + \mu v_2 \mid \tau(w, \gamma) = (1 + v_1)\tau(w^0, \gamma^0), \quad \kappa(w, \gamma) = (1 + v_2)\kappa(w^0, \gamma^0). \tag{23}$$

Problem (23) uses a compact notation highlighting the changes compared to (20) in Table 3 and Table 4. As in Table 3 and Table 4, w depends on $\beta \in [0,5]$ and γ depends on $\alpha \in [0,2]$, $c \in [0.1,1]$, and $\delta \in [0,1]$, and w^0, γ^0 refer to $\alpha = \beta = \delta = 0$, and $c = 1$. The restriction $v_1, v_2 \leq 0.1$ implies that neither τ nor κ are allowed to be more than 10% worse than the choice w^0, γ^0. Again, $\mu > 0$ is a weight balancing optimization error and stochastic error; however, due to the upper bound on v_1, v_2 the dependence on μ of the optimal solution turns out to be less pronounced.

Using "min_fc.m" from [7] with different starting points, approximate local optimal solutions were computed for Problem (23). For upper bounds $v_1, v_2 \leq 0$ aiming at a simultaneous reduction of both τ and κ, no point apart from $\alpha = \beta = 0$ could be identified via "min_fc.m". However, allowing a small slack of $v_1, v_2 \leq 0.1$ as in (23) led to

$$\alpha = \delta = 0, \quad \beta = 0.7116, \quad \text{and} \quad c = 1 \tag{24}$$

for a "generic" situation with $D_{1,1} = 0.03$, $D_{n,n} = 1$ and $k^{max} = 10000$. For these values of α, β, c, δ, the value of κ did increase by 10% (from 0.3325 to 0.3658) while τ decreased by 97.4% (from 3.2e-3 to 8.7e-5) compared to the choice $\alpha = \beta = \delta = 0$ and $C = 1$. The large value of κ indicates for this condition number that 10^4 sgd-iterations with noise do not reduce the expected error below approximately $1/3$ of the size of the noise-terms. If k^{max} is increased to 10^6, the value of κ reduces to 0.0366 and τ reduces to $3.3 \cdot 10^{-8}$.

For large initial errors $\|x^0\|$ this reduction of τ will outweigh the 10%-increase of κ. In this situation a significant gain in the optimization error could be achieved when allowing a small increase of the stochastic error.

To test the robustness of this solution, the other parameters k^{max} and $D_{1,1}$ of Table 1 and Table 2 were evaluated as well for $\alpha = \delta = 0$, $\beta = 0.7116$: For higher condition numbers up to 10^4 the increase of κ is less than 21% and it is less than 10% for smaller condition numbers or larger values of k^{max}. Likewise, the reduction of τ deteriorates for increasing condition numbers but improves for smaller condition numbers or for larger values of k^{max}.

3.1.1 Practical Parameters

Summarizing, while a step length reduction generally does not lead to a significant improvement of convergence, moderately growing weights such as $w_j = j^{0.7}$ do lead to faster reduction of the optimization error without deteriorating the stochastic error. Thus, the parameter setting (24) is used in the numerical examples in Subsection 3.2 with the slight modification that β is set to $\beta = 0.7$ (for simplicity).

3.2 Test Examples

Algorithm 1 is rather simple and the results in Section 2. are not very strong but "robust" in the sense that they are independent of the dimension n and of the number m in the definition (1) of f.

3.2.1 Simple Examples

For the first set of examples a situation is considered that does satisfy the strong assumptions of Section 2. and where $m = \infty$ so that variance reduction by periodic full gradient evaluations is not possible.

For a given dimension $n \in [10^2, 10^8]$ a non-singular randomly generated diagonal matrix $D \in \mathbb{R}^{n \times n}$ is chosen with diagonal entries in $[0.1, 1]$ and a scaling factor $\rho := 1/\sqrt{n}$. For $k = 1, 2, \ldots$ random vectors b^k are drawn independently from an n-variate normal distribution with expected value 0_n and covariance matrix $\rho^2 I_n$, i.e. $b^k \sim N_n(0_n, \rho^2 I_n)$. Thus, $E(\|b^k\|_2^2) = n\rho^2 = 1$ independently of the dimension n. The functions f_k are then given by

$$f_k(x) \equiv \tfrac{1}{2} x^{\mathrm{T}} D x + (b^k)^{\mathrm{T}} x$$

with $E(f_k(x)) = \tfrac{1}{2} x^{\mathrm{T}} D x$.

For $k^{max} = 10^5$ the results of Algorithm 1 with $\alpha = 0$, $\beta = 0.7$ and $x^0 = e/\sqrt{n}$ (i.e. $\|x^0\|_2 = 1$) are listed for different dimensions:

n	10^1	10^2	10^3	10^4	10^5
$\|\bar{x}^{final}\|_2$	0.0123	0.0101	0.0129	0.0135	0.0133

Table 5: Final error in dependence of the dimension.

For the examples listed above the number of iterations indeed does not display any dependence on the dimension.

For the same setting with $n = 100$ and different values of k^{max} the results of Algorithm 1 are as follows:

k^{max}	10^4	10^5	10^6	10^7	10^8
$\|\bar{x}^{final}\|_2$	3.9e-2	1.4e-2	3.8e-3	1.2e-3	3.9e-4

Table 6: Final error in dependence of k^{max}.

Here, the convergence is rather slow with a growing number of iterations – it is of the order $1/\sqrt{k^{max}}$, i.e., as in the analysis of Polyak and Juditsky [14], the stochastic effects do dominate the convergence. The reductions of the initial error observed in Table 6 are better by a factor of 2.5 or 2.6 than the theoretical bounds $\kappa(0, 0)$ listed in the second column of Table 2.

For all test runs listed in Table 5 and Table 6, the final iterate had a norm $\|x^{k^{max}}\| \in [1.21, 1.39]$ — since $\alpha = 0$, the step length was constant and the final iterate was largely determined by the noise.

The same setting with $n = 100$ and $k^{max} = 10^5$ was now applied to different starting points $x^0 = \lambda e/\sqrt{n}$ with $\lambda > 0$ and with $\beta = 0.7$ as well as $\beta = 0$.

16

λ (i.e. $\|x^0\|_2$)		10^0	10^2	10^4	10^6	10^8
$\|\bar{x}^{final}\|_2$	$\beta = 0$	1.0e-2	1.1e-2	2.9e-1	28.6	3179
	$\beta = 0.7$	1.4e-2	1.2e-2	1.3e-2	5.4e-2	6.1

Table 7: Final error in dependence of $\|x^0\|_2$.

For a large initial error λ there is a significant gain when choosing $\beta = 0.7$ compared to $\beta = 0$ while there is not much loss of the choice $\beta = 0.7$ for small initial errors. (For $\lambda = 10^8$ the final iterate x^{100000} has norm 1.26, i.e. it is much closer to the optimal solution than the average $\|\bar{x}^{final}\|_2$ with $\beta = 0$.)

For large initial errors, the value of τ will dominate the convergence behavior. Indeed, for the entry with $\beta = 0.7$ in the last column of Table 7 a reduction of the initial error by $10^8/6.1 \approx 1.6 \cdot 10^7$ can be observed which is larger by a (moderate) factor of about 3.5 than the theoretical bound $1/\tau(0, 0.7) \approx 4.5 \cdot 10^6$ and close to bound given by the last entry of Column 2 in Table 1 with $1/\tau(0, 0) = 10^{7.0458} \approx 1.1 \cdot 10^7$ for $\beta = 0$.

Summarizing, the above simple examples confirm the independence of the theoretical results with respect to the dimension n and the number m of functions in the definition of f, as long as the variance $\|\nabla f(x) - \nabla f_k(x)\|^2$ is bounded independent of x and n. The observed results also indicate that the theoretical results in Table 1 and Table 2 are not overly pessimistic.

3.2.2 Relaxing the Assumptions

To test the limits of Algorithm 1 the next example concerns a somewhat more difficult case where again $m = \infty$, and where each single function f_k carries rather little information about the function f to be minimized.

For this set of examples a non-singular randomly generated fixed matrix $A \in \mathbb{R}^{n \times n}$ is chosen and a scaling factor $\rho := 1/\sqrt{n}$. For $k = 1, 2, \ldots$ random vectors r^k and b^k are drawn independently from n-variate normal distributions with expected value 0_n and covariance matrices I_n respectively $\rho^2 I_n$, i.e. $r^k \sim N_n(0_n, I_n)$ and $b^k \sim N_n(0_n, \rho^2 I_n)$. The choice of ρ implies that $E(\|b^k\|_2^2) = 1$. Define $a^k := Ar^k$ (so that $a^k \sim N_n(0_n, AA^T)$) and for given $x \in \mathbb{R}^n$ let $f_k(x) := \frac{1}{2}((a^k)^T x)^2 + (b^k)^T x$ with the expected value

$$
\begin{aligned}
E\left(f_k(x)\right) &= E\left(\frac{1}{2}((a^k)^T x)^2 + (b^k)^T x\right) \\
&= \frac{1}{2}E\left(x^T a^k (a^k)^T x\right) + \underbrace{E\left(b^k\right)^T}_{=0_n} x \\
&= \frac{1}{2}x^T E\left(a^k (a^k)^T\right) x,
\end{aligned}
$$

17

where the last equation uses the linearity of the expected value for fixed x. Since

$$E\left(a^k(a^k)^{\mathrm{T}}\right) = E\left(Ar^k(Ar^k)^{\mathrm{T}}\right) = E\left(Ar^k(r^k)^{\mathrm{T}}A^{\mathrm{T}}\right) = AE\left(r^k(r^k)^{\mathrm{T}}\right)A^{\mathrm{T}} = AA^{\mathrm{T}},$$

one can proceed

$$E\left(f_k(x)\right) = \tfrac{1}{2}x^{\mathrm{T}}AA^{\mathrm{T}}x =: f(x).$$

And since the fourth momenta of r^k and b^k exist and f_k is a quadratic function of r^k and b^k it follows that f_k has bounded variance and

$$\lim_{m\to\infty} \frac{1}{m}\sum_{k=1}^{m} f_k(x) = f(x)$$

exists almost surely. Moreover, $\nabla f(x) = AA^{\mathrm{T}}x$, which coincides with the expected value of $\nabla f_k(x)$:

$$E(\nabla f_k(x)) = E\left(a^k(a^k)^{\mathrm{T}}\right)x + E\left(b^k\right) = AA^{\mathrm{T}}x = \nabla f(x).$$

The noise defined by $\boldsymbol{\xi}^k := \nabla f_k(x) - \nabla f(x) = a^k(a^k)^{\mathrm{T}}x + b^k - AA^{\mathrm{T}}x$ has expected value

$$E(\boldsymbol{\xi}^k) := E(\nabla f_k(x)) - \nabla f(x) = 0_n$$

and the covariance matrix is given by

$$E(\boldsymbol{\xi}^k(\boldsymbol{\xi}^k)^{\mathrm{T}}) = \rho^2 I + AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}} + \|A^{\mathrm{T}}x\|_2^2 AA^{\mathrm{T}}. \tag{25}$$

To keep the presentation self-contained a short proof of (25) is given in the appendix.

If $x \to 0$, the noise terms $\boldsymbol{\xi}^k$ indeed are i.i.d. in the limit, as assumed in (4), but for larger starting errors $\|x^0\|$ this assumption is violated, more so when the dimension n grows large.

Note that (since the matrices on the right hand side of (25) all are positive semidefinite)

$$\|E(\boldsymbol{\xi}^k(\boldsymbol{\xi}^k)^{\mathrm{T}})\|^{1/2} \geq \rho \max\left\{1, \|AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}}\|^{1/2}, \sqrt{n}\,\|A\,\mathrm{Diag}((A^{\mathrm{T}}x)^2)A^{\mathrm{T}}\|^{1/2}\right\}$$

where $\|AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}}\|^{1/2} = \|\nabla f(x)\|$. Hence, (25) implies that standard deviation of the noise $\boldsymbol{\xi}^k$ (that is added to the gradient at each iteration of Algorithm 1) always dominates the norm of the gradient itself.

Similar to the analysis in Section 2., again, for the numerical experiments it is sufficient to use a diagonal matrix D in place of A. Indeed, consider the singular value decomposition $A = UDV^{\mathrm{T}}$ with a diagonal matrix D containing the singular values of A and orthogonal matrices U, V. Since $a^k = Ar^k = UDV^{\mathrm{T}}r^k \sim N_n(0_n, UDDU^{\mathrm{T}})$ one obtains

$$E\left(f_k(x)\right) = \tfrac{1}{2}x^{\mathrm{T}}E\left(a^k(a^k)^{\mathrm{T}}\right)x = \tfrac{1}{2}x^{\mathrm{T}}UDDU^{\mathrm{T}}x = \tfrac{1}{2}z^{\mathrm{T}}DDz := \tilde{f}(z)$$

with the transformation $z := U^{\mathrm{T}}x$. Likewise, (25) translates to an equivalent formula for z and $\tilde{\boldsymbol{\xi}}^k := U^{\mathrm{T}}\boldsymbol{\xi}^k$.

The examples in Table 8 and Table 9 refer to $n = 100$, randomly generated D (uniform distribution scaled to $D_{n,n} = n^{-1/2}$ and $D_{1,1} = (10n)^{-1/2}$) so that the condition number of D^2 is 10. The scaling by $n^{-1/2}$ was chosen to compensate for the norm $\|r^k\|$ which is of the order $n^{1/2}$. For the choice $k^{max} = 1000$ and $\beta = 0.7$ (for comparison also $\beta = 0$) the following average final errors were obtained:

$\|x^0\|_2$		10^{-1}	10^0	10^1	10^2	10^3	10^4
$\|\bar{x}^{final}\|_2$	$\beta = 0$	0.08	0.32	3.3	33	351	3218
	$\beta = 0.7$	0.09	0.30	2.6	24	266	2817

Table 8: Final error in dependence of $\|x^0\|$ when $k^{max} = 1000$.

When $\|x^0\|_2$ is small, the final iterate is mostly determined by the noise and it may occur that $\|\bar{x}^{final}\|_2 \gg \|x^0\|_2$. For the same setting as above and $\|x^0\|_2 = 10^2$ different values of k^{max} led to the following results (again $\beta = 0$ is listed for comparison):

k^{max}		10^3	10^4	10^5	10^6	10^7	10^8
$\|\bar{x}^{final}\|_2$	$\beta = 0$	37	4.6	0.41	0.040	0.0049	6.0e-4
	$\beta = 0.7$	23	1.1	0.022	0.0048	0.0018	4.6e-4

Table 9: Final error in dependence of k^{max} when $\|x^0\|_2 = 10^3$.

Above, it takes about 10 times longer for the algorithm with $\beta = 0$ to reach an error of 0.0048 or 0.0049 than it takes with the choice $\beta = 0.7$.

When multiplying D with a positive constant greater than one, convergence actually improves (as the noise ratio $\|b^k\|/\|a^k\|$ decreases), but overall, this is an example where the stochastic effects dominate. This also implies that it may be difficult to improve over the simple scheme of Algorithm 1 for this type of example.

3.2.3 Non-Quadratic Test Examples

The intent of this paper of course is to motivate a rule that is more generally applicable for stochastic gradient descent approaches. This motivation is supported by the observation that if the functions f_i are convex and twice continuously differentiable, then locally, the functions can be closely approximated by quadratic functions for which the analysis holds.

The MNIST database [3] provides 70000 handwritten digit images with corresponding labels 0,1,...,9. Each image consists of $n = 28 \cdot 28 = 784$ pixels with entries between 0 and 1. To test the performance of Algorithm 1, $m = 60000$ of the labeled images are used as training data to find a rule that predicts whether the digits of the remaining 10000 test images are 0 or not. This prediction can be compared with the labels of the test images to determine the false classification rate "FCR" i.e. the percentage of incorrectly classified

images.

For the numerical experiments the data for the i-th image is put into the following form: the vector $a_i \in [0,1]^{784}$ contains the information about the image itself and $b_i \in \{-1,+1\}$ is the corresponding label, where "+1" means that the image shows the digit 0 and "-1" means this is not the case.

Let $\sigma: \mathbb{R} \to]0,1[$, $\sigma(t) := 1/(1 + \mathrm{e}^{-t})$ be the logistic function with derivative $\sigma'(t) = \sigma(t)(1 - \sigma(t))$ and consider $f: \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x) := -\frac{1}{m} \sum_{i=1}^{m} \log(\sigma(b_i(a_i^{\mathrm{T}} x))).$$

Note that $t \mapsto \log(\sigma(t))$ is a smooth (and asymptotically exact) approximation of the function $t \mapsto \max\{t, 0\}$. Thus minimizing f, approximates the maximization of the terms $\max\{b_i(a_i^{\mathrm{T}} x), 0\}$ for $1 \le i \le m$. If all terms $b_i(a_i^{\mathrm{T}} x)$ are positive then

$$a_i^{\mathrm{T}} x > 0 \quad \text{for all } i \text{ with } b_i = 1 \quad \text{and} \quad a_i^{\mathrm{T}} x < 0 \quad \text{for all } i \text{ with } b_i = -1. \tag{26}$$

This motivates the following classification rule: Let an approximate minimizer $\bar{x}^{final}$ of f be given and a test image a_{new}. Then a_{new} is classified to represent the digit "0" if $a_{\mathrm{new}}^{\mathrm{T}} \bar{x}^{final} > 0$, and a_{new} is classified not to represent the digit "0" otherwise.

The derivatives of f are given by

$$\nabla f(x) = -\frac{1}{m} \sum_{i=1}^{m} b_i a_i \frac{\sigma'(b_i(a_i^{\mathrm{T}} x))}{\sigma(b_i(a_i^{\mathrm{T}} x))} = -\frac{1}{m} \sum_{i=1}^{m} b_i a_i (1 - \sigma(b_i(a_i^{\mathrm{T}} x))),$$

$$\nabla^2 f(x) = -\frac{1}{m} \sum_{i=1}^{m} b_i a_i (-b_i a_i^{\mathrm{T}} \sigma'(b_i(a_i^{\mathrm{T}} x)))$$

$$= \underbrace{\frac{1}{m}}_{>0} \sum_{i=1}^{m} \underbrace{b_i^2}_{=1} \underbrace{a_i a_i^{\mathrm{T}}}_{\succeq 0} \underbrace{\sigma(b_i(a_i^{\mathrm{T}} x))(1 - \sigma(b_i(a_i^{\mathrm{T}} x)))}_{>0} \succeq 0.$$

Here, f is convex since the Hessian matrix is positive semidefinite, i.e. $\nabla^2 f(x) \succeq 0$.

To estimate a value for the maximum step length c in Algorithm 1, consider the case that a_i is a random vector whose components are continuously uniformly distributed on $]0,1[$. Thus, every component of a_i has expected value $1/2$ and second momentum $1/3$ and the components of a_i are independent of each other. Since

$$E(a_i a_i^{\mathrm{T}})_{r,s} = \begin{cases} 1/3 & \text{if } r = s \\ 1/4 & \text{if } r \ne s \end{cases}$$

the Hessian matrix can be estimated by

$$E(\nabla^2 f(0)) = \frac{1}{m} \sum_{i=1}^{m} E(a_i a_i^{\mathrm{T}}) \tfrac{1}{2}(1 - \tfrac{1}{2}) = \frac{1}{4m} m \left(\tfrac{1}{4} \mathbb{1}_{n \times n} + \tfrac{1}{12} I_n \right) = \tfrac{1}{16} \mathbb{1}_{n \times n} + \tfrac{1}{48} I_n$$

with the maximum eigenvalue $n/16 + 1/48$. This leads to the approximation $c := 16/n$ used for the results in Table 10.

In Table 10 the results of Algorithm 1 with $\alpha = 0$ and $\beta \in \{0, 0.7\}$ are listed for the initial value $w_0 = 0_{784}$ and for varying numbers of iterations k^{max}:

k^{max}		10^3	10^4	10^5	10^6	10^7
$\|\nabla f(\bar{x}^{final})\|_2$	$\beta = 0$	6.7e-2	1.1e-2	2.4e-3	1.2e-3	7.9e-4
	$\beta = 0.7$	5.0e-2	8.3e-3	1.8e-3	7.6e-4	4.1e-4
FCR	$\beta = 0$	1.87%	0.97%	0.74%	0.62%	0.70%
	$\beta = 0.7$	1.70%	0.96%	0.73%	0.63%	0.70%

Table 10: Norm of the gradient and false classification rate in dependence of k^{max}.

For this example it was also possible to apply Newton's method with line search which generated an approximation x^{opt} with $\|\nabla f(x^{opt})\|_2 \approx 2.3 \cdot 10^{-10}$ and an associated false classification rate of FCR$\approx 0.80\%$. The Hessian of f at x^{opt} was numerically singular and the spectrum of $\nabla^2 f(x^{opt})$ was quite dense near zero and did not allow a clear identification of the null space. It was not possible to give a reliable estimate for the condition number of the Hessian of f, even when restricted to the range space of $\nabla^2 f(x^{opt})$. The Hessian certainly was very far from being well-conditioned.

For all approximate solutions $\bar{x}^{final}$ generated in Table 10, the distance $\|\bar{x}^{final} - x^{opt}\|$ was about the same, namely close to 1100. In particular, a convergence of the iterates of Algorithm 1 to a minimizer of f could not be observed. Nevertheless a small gain in the rate of convergence with $\beta = 0.7$ compared to $\beta = 0$ could be observed for smaller values of k^{max}. For larger values of k^{max} the asymptotic optimality of $\beta = 0$ in [14] can be confirmed in the sense that there is not much difference of $\beta = 0.7$ and $\beta = 0$.

It is stressed that the intention of this example was not to propose a new classification scheme for MNIST – other classification schemes are certainly better – but to test Algorithm 1 with a somewhat realistic example. In particular, the deterioration of the false classification rate in the last column – and for x^{opt} – indicate that the phenomenon of overfitting must be addressed with this approach.

4. Appendix

Proof of (25):
Note that

$$E(\boldsymbol{\xi}^k(\boldsymbol{\xi}^k)^{\mathrm{T}}) = E((\nabla f_k(x) - \nabla f(x))(\nabla f_k(x) - \nabla f(x))^{\mathrm{T}})$$
$$= E((aa^{\mathrm{T}}x + b - AA^{\mathrm{T}}x)(aa^{\mathrm{T}}x + b - AA^{\mathrm{T}}x)^{\mathrm{T}})$$

where $a = a^k = Ar^k$ and $b = b^k$ with independent normally distributed vectors $r^k \sim N_n(0_n, I)$ and $b^k \sim N_n(0_n, \rho^2 I)$, a fixed matrix A and a fixed vector x. Multiplying the product in the

expected value returns a sum of 9 terms that are considered one by one:

The expectation of the constant term $AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}}$, of course, is the term itself.

The expectations of $-AA^{\mathrm{T}}xb^{\mathrm{T}}$ and $-bx^{\mathrm{T}}AA^{\mathrm{T}}$ are both zero (since $AA^{\mathrm{T}}x$ is a fixed vector).

The expectation of bb^{T} is $\rho^2 I$.

The expectation of the two terms $-AA^{\mathrm{T}}xx^{\mathrm{T}}aa^{\mathrm{T}}$ and $-aa^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}}$ is $-AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}}$ each.

The expectation of $aa^{\mathrm{T}}xx^{\mathrm{T}}aa^{\mathrm{T}}$ is given by

$$E(aa^{\mathrm{T}}xx^{\mathrm{T}}aa^{\mathrm{T}}) = E(A\,rr^{\mathrm{T}}\underbrace{A^{\mathrm{T}}xx^{\mathrm{T}}A}_{=:B}\,rr^{\mathrm{T}}A^{\mathrm{T}}) = A\,E(rr^{\mathrm{T}}Brr^{\mathrm{T}})\,A^{\mathrm{T}}$$

where the i-th component of $r = r^k \sim N_n(0_n, I_n)$, denoted by $r_i \sim N(0,1)$, has the momenta $E(r_i) = 0$, $E(r_i^2) = 1$ and $E(r_i^4) = 3$. By distinguishing all possible cases one obtains

$$E(r_i r_j r_p r_q) = \begin{cases} 3 & \text{if } i = j = p = q, \\ 1 & \text{if } i = j \neq p = q, \\ 1 & \text{if } i \neq j \text{ and } ((i = p,\ j = q) \text{ or } (i = q,\ j = p)), \\ 0 & \text{else.} \end{cases}$$

. With $i, j \in \{1, \ldots, n\}$ the expected value of $rr^{\mathrm{T}}Brr^{\mathrm{T}}$ is given componentwise by

$$E(rr^{\mathrm{T}}Brr^{\mathrm{T}})_{i,j} = E(e_i^{\mathrm{T}}rr^{\mathrm{T}}Brr^{\mathrm{T}}e_j) = E(r_i r_j r^{\mathrm{T}}Br) = E\left(r_i r_j \sum_{p,q} B_{p,q} r_p r_q\right)$$

$$= \sum_{p,q} B_{p,q} E(r_i r_j r_p r_q) = \begin{cases} 3B_{i,i} + \sum_{p=1, p \neq i}^{n} B_{p,p} & \text{if } i = j \\ 2B_{i,j} & \text{else} \end{cases}$$

$$= \begin{cases} 2B_{i,i} + \mathrm{tr}(B) & \text{if } i = j \\ 2B_{i,j} & \text{else.} \end{cases}$$

Consequently, one obtains

$$E(aa^{\mathrm{T}}xx^{\mathrm{T}}aa^{\mathrm{T}}) = A\,E(rr^{\mathrm{T}}Brr^{\mathrm{T}})\,A^{\mathrm{T}}$$
$$= A\,(2B + \mathrm{tr}(B)I_n)\,A^{\mathrm{T}}$$
$$= 2AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}} + AA^{\mathrm{T}}\mathrm{tr}(A^{\mathrm{T}}xx^{\mathrm{T}}A)$$
$$= 2AA^{\mathrm{T}}xx^{\mathrm{T}}AA^{\mathrm{T}} + \|A^{\mathrm{T}}x\|_2^2 AA^{\mathrm{T}}.$$

Finally, the expectations of $bx^{\mathrm{T}}aa^{\mathrm{T}}$ and of $aa^{\mathrm{T}}xb^{\mathrm{T}}$ both are zero, since b is chosen independently of a and thus also of $x^{\mathrm{T}}aa^{\mathrm{T}}$.

Summing up all 9 expectations above leads to (25).

5. Conclusion

The optimization error and the stochastic error are analyzed for the simple case of a stochastic gradient method applied to a strongly convex quadratic function with stochastic gradients

that satisfy the assumption of being i.i.d. By optimizing both errors for an algorithm with finite termination it was possible to modify the known asymptotically optimal parameter selection of a stochastic gradient method. As predicted by the analysis in [14], for large numbers of iterations k^{max} the results cannot be improved, but for moderate numbers of iterations the numerical experiments confirm a gain in accuracy. This gain can be achieved by a simple parameter selection and without additional computational cost.

The extension to smooth non-convex functions or to limited memory Quasi-Newton approaches such as presented in [1] are the subject of future research.

References

[1] BURDAKOV, Oleg ; GONG, Lujin ; ZIKRIN, Spartak ; YUAN, Ya-xiang: On efficiently combining limited-memory and trust-region techniques. In: *Mathematical Programming Computation* 9 (2017), S. 101–134

[2] COHEN, Kobi ; NEDIĆ, Angelia ; SRIKANT, R: On projected stochastic gradient descent algorithm with weighted averaging for least squares regression. In: *IEEE Transactions on Automatic Control* 62 (2017), Nr. 11, S. 5974–5981

[3] DENG, Li: The mnist database of handwritten digit images for machine learning research. In: *IEEE Signal Processing Magazine* 29 (2012), Nr. 6, S. 141–142

[4] FRIEDMAN, Jerome H.: Stochastic gradient boosting. In: *Computational Statistics and Data Analysis* 38 (2002), S. 367–378

[5] GUO, Hao ; JIN, Jiyong ; LIU, Bin: Stochastic weight averaging revisited. In: *arXiv preprint arXiv:2201.00519* (2022)

[6] IZMAILOV, Pavel ; PODOPRIKHIN, Dmitrii ; GARIPOV, Timur ; VETROV, Dmitry ; WILSON, Andrew G.: Averaging weights leads to wider optima and better generalization. In: *arXiv preprint arXiv:1803.05407* (2018)

[7] JARRE, Florian ; LIEDER, Felix: A Derivative-Free and Ready-to-Use NLP Solver for Matlab or Octave. In: *Preprint, http://www.opt.uni-duesseldorf.de/~jarre/dot/mwdc.pdf* (2017)

[8] JOHNSON, Rie ; ZHANG, Tong: Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In: BURGES, C.J. (Hrsg.) ; BOTTOU, L. (Hrsg.) ; WELLING, M. (Hrsg.) ; GHAHRAMANI, Z. (Hrsg.) ; WEINBERGER, K.Q. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 26, Curran Associates, Inc., 2013

[9] LAZAR, Markus ; JARRE, Florian: Calibration by Optimization Without Using Derivatives. In: *Optimization and Engineering* 17 (2016)

[10] LIU, Jun ; YUAN, Ye: On Almost Sure Convergence Rates of Stochastic Gradient Methods. In: *arXiv preprint arXiv:2202.04295* (2022)

[11] LOIZOU, Nicolas ; RICHTARIK, Peter: Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. In: *Computational Optimization and Applications* 77 (2020), S. 653–710

[12] NEEDELL, Deanna ; WARD, Rachel ; SREBRO, Nati: Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In: *Advances in neural information processing systems* 27 (2014)

[13] NEU, Gergely ; ROSASCO, Lorenzo: Iterate averaging as regularization for stochastic gradient descent. In: *Conference On Learning Theory* PMLR, 2018, S. 3222–3242

[14] POLYAK, Boris T. ; JUDITSKY, Anatoli B.: Acceleration of stochastic approximation by averaging. In: *SIAM journal on control and optimization* 30 (1992), Nr. 4, S. 838–855

[15] SEBBOUH, Othmane ; GOWER, Robert M. ; DEFAZIO, Aaron: Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In: *Conference on Learning Theory* PMLR, 2021, S. 3935–3971

[16] SHAMIR, Ohad ; ZHANG, Tong: Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In: *International conference on machine learning* PMLR, 2013, S. 71–79