ON THE INFLUENCE OF MOMENTUM ACCELERATION ON ONLINE LEARNING

Kun Yuan, Bicheng Ying and Ali H. Sayed

Department of Electrical Engineering
University of California, Los Angeles

ABSTRACT

This paper examines the convergence rate and mean-square-error performance of momentum stochastic gradient methods in the constant step-size and slow adaptation regime. The results establish that momentum methods are equivalent to the standard stochastic gradient method with a re-scaled (larger) step-size value. The equivalence result is established for *all* time instants and not only in steady-state. The analysis is carried out for general risk functions, and is not limited to quadratic risks. One notable conclusion is that the well-known benefits of momentum constructions for deterministic optimization problems do not necessarily carry over to the stochastic setting when gradient noise is present and continuous adaptation is necessary. The analysis suggests a method to enhance performance in the stochastic setting by tuning the momentum parameter over time.

Index Terms— stochastic gradient descent, momentum acceleration, heavy-ball method, Nesterov method.

1. INTRODUCTION AND RELATED WORKS

Stochastic optimization focuses on the problem of optimizing the expectation of a loss function, written as

$$\min_{w\in\mathbb{R}^M} \; J(w) \;\stackrel{\Delta}{=}\; \mathbb{E}_{\boldsymbol{\theta}}[Q(w;\boldsymbol{\theta})], \tag{1}$$

where $\boldsymbol{\theta}$ is a random variable whose distribution is generally unknown and $J(w)$ is a convex function (usually strongly-convex due to regularization). Problems of this kind are common in many contexts, including in several adaptation and machine learning formulations [1–4].

When $J(w)$ is differentiable, one of the most popular techniques to seek minimizers for (1) is to employ the *stochastic* gradient method, which takes the form:

$$\boldsymbol{w}_i \;=\; \boldsymbol{w}_{i-1} \;-\; \mu\nabla_w\, Q(\boldsymbol{w}_{i-1};\boldsymbol{\theta}_i), \quad i \ge 0, \tag{2}$$

where $\mu > 0$ is a step-size parameter, $\boldsymbol{\theta}_i$ is the observation of $\boldsymbol{\theta}$ at iteration i, and $\nabla_w Q(\cdot)$ denotes the gradient vector of the loss function relative to w. Note that we use boldface letters to refer to random quantities. In this paper we focus on constant step-size implementations for two main reasons. First, they endow the resulting recursions with continuous adaptation, learning, and tracking abilities and, second, they help attain exponential convergence rates in the order of $O(\alpha^i)$ for some $\alpha \in (0,1)$. This is in contrast to the slower rate of $O(1/i)$ that is afforded by decaying step-sizes [5–8]. Although constant step-size implementations can cause small deterioration in the limiting accuracy of the iterates, this deterioration is tolerable in most large-scale learning and adaptation problems [9, 10].

Now, seeking the minimizer(s) of problems of type (1) is challenging because the cost function, $J(w)$, is generally unknown due to the lack of information about the probability distribution of the data. This *stochastic* optimization problem is in contrast to *deterministic* problems where the cost function, $J(w)$, is known and, therefore, its gradient vector is also known and can be used in (2) in place of the gradient vector of the loss function. The resulting gradient-descent techniques are efficient and have lower computational demands than more sophisticated methods (say, of the Newton type). Nevertheless, they tend to exhibit slow convergence rates. Several useful methods have been proposed in the literature to speed up the convergence of gradient-descent recursions for deterministic optimization problems. Among these methods, the heavy-ball technique [6, 11] and Nesterov's acceleration [12–14] are the most successful variations. Both methods rely on the addition of a momentum term to the recursion and it is known that, when the risk function $J(w)$ is strongly convex and has Lipschitz continuous gradients, these methods succeed in attaining the optimal exponential convergence rate. There are also other advantages for risk functions that are not necessarily strongly-convex.

Motivated by these useful acceleration properties in the *deterministic* context, momentum terms have been subsequently introduced into *stochastic* optimization algorithms as well [6, 15–21] and applied, for example, to problems involving the tracking of chirped sinusoidal signals [22] or deep learning [23]. However, the analysis in this paper will show that their advantages from deterministic optimization do not necessarily carry over to the stochastic setting due to the presence of gradient noise (which is the difference between the actual gradient vector and its approximation). Specifically, we will show that any advantage they bring forth can be achieved by staying with the original stochastic-gradient algorithm and adjusting its step-size to a larger value. For instance, for optimization problem (1), we will show that if the step-sizes, μ_m for the momentum (heavy-ball or Nesterov) methods and μ for the standard stochastic gradient algorithms, are sufficiently small and satisfy the relation

$$\mu = \frac{\mu_m}{1-\beta} \tag{3}$$

where $\beta \in [0,1)$ is the coefficient of the momentum term, then it will hold that

$$\mathbb{E}\|\boldsymbol{w}_{m,i} - \boldsymbol{w}_i\|^2 = O(\mu^{3/2}),\; i = 0,1,2,\ldots \tag{4}$$

This work was supported in part by NSF grants CIF-1524250 and ECCS-1407712, DARPA project N66001–14–2–4029, and by a Visiting Professorship from the Leverhulme Trust, United Kingdom. Email:{kunyuan,ybc,sayed}@ucla.edu

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

where $\boldsymbol{w}_{m,i}$ and $\boldsymbol{w}_i$ denote the iterates generated at time i by the momentum and standard implementations, respectively. In the special case when $J(w)$ is quadratic in w, as happens in mean-square-error design problems, we can tighten (4) to

$$\mathbb{E}\|\boldsymbol{w}_{m,i} - \boldsymbol{w}_i\|^2 = O(\mu^2),\ i = 0, 1, 2, \dots \tag{5}$$

What is important to note is that, we will show that these results hold *for every* i, and not only asymptotically. Therefore, when μ is sufficiently small, property (4) establishes that the stochastic gradient method and the momentum versions are fundamentally equivalent since their iterates evolve close to each other at all times. The analysis in later sections will further suggest a technique to recover the faster performance of the momentum implementations by tuning the momentum parameter over time.

1.1. Related Works in the Literature

There exist useful results in the literature that relate to special instances of the general framework developed in this work, mainly for the mean-square-error case when $J(w)$ is quadratic in w. We do not limit our analysis to this case and our results are applicable to a broader class of problems beyond mean-square-error estimation (e.g., logistic regression is covered). The treatment of the general $J(w)$ case is demanding because the Hessian matrix of $J(w)$ is now a matrix-function and is $w-$dependent, whereas it is a constant in the quadratic case.

References [6, 15, 16] studied the heavy-ball stochastic gradient method for quadratic costs. They observed that although the heavy-ball method increases the convergence rate, it nevertheless results in larger misadjustment in steady-state. References [17, 18] studied heavy-ball LMS closely and claimed that no significant gain is achieved in convergence speed if both the heavy-ball and standard LMS algorithms are tuned to have similar *steady-state* mean-square-deviation (MSD) performance. Reference [19] observed that when the step-sizes satisfy relation (3), then the heavy-ball LMS algorithm is "equivalent" to standard LMS. However, the notion of "equivalence" in this work is only referring to the fact that the algorithms have similar starting convergence rates and similar steady-state MSD. There was no analysis of the behavior of the algorithms during all stages of learning – see also [20]. Another useful work is [21], which considered the heavy-ball stochastic gradient method for general risks, $J(w)$. This work concluded that heavy-ball can be equivalent to the standard stochastic gradient method asymptotically (i.e., for i large enough). All of these works were limited to the heavy-ball technique; they did not examine Nesterov's technique, which is also applicable to stochastic gradient learning – see, e.g., [23].

2. MOMENTUM ACCELERATION

The analysis in this work is carried out under the following assumption on $J(w)$, which is common in the context of adaptation and learning and is often automatically satisfied due to the use of regularization terms. The condition essentially amounts to assuming that $J(w)$ is strongly-convex with Lipschitz gradient.

Assumption 1 (Conditions on risk function). *The cost function $J(w)$ is twice-differentiable and its Hessian matrix satisfies*

$$0 < \nu I_M \le \nabla^2 J(w) \le \delta I_M, \tag{6}$$

for some positive parameters $\nu \le \delta$. ■

We denote the minimizer for problem (1) by w^o. Under Assumption 1, this minimizer is unique. We carry out the analysis by considering the following general form of a stochastic-gradient implementation, with two momentum parameters $\beta_1, \beta_2 \in [0, 1)$:

$$\boldsymbol{\psi}_{i-1} = \boldsymbol{w}_{i-1} + \beta_1(\boldsymbol{w}_{i-1} - \boldsymbol{w}_{i-2}), \tag{7}$$

$$\boldsymbol{w}_i = \boldsymbol{\psi}_{i-1} - \mu_m \nabla_w Q(\boldsymbol{\psi}_{i-1}; \boldsymbol{\theta}_i) + \beta_2(\boldsymbol{\psi}_{i-1} - \boldsymbol{\psi}_{i-2}), \tag{8}$$

with initial conditions

$$\boldsymbol{w}_{-2} = \boldsymbol{\psi}_{-2} = \text{initial states}, \tag{9}$$

$$\boldsymbol{w}_{-1} = \boldsymbol{w}_{-2} - \mu_m \nabla_w Q(\boldsymbol{w}_{-2}; \boldsymbol{\theta}_{-1}). \tag{10}$$

We refer to this formulation as the momentum stochastic gradient method. When $\beta_1 = 0$ and $\beta_2 = \beta$ we recover the heavy-ball algorithm [6, 11], and when $\beta_2 = 0$ and $\beta_1 = \beta$, we recover Nesterov's algorithm [13]. These two situations correspond to the condition:

$$\beta_1 + \beta_2 = \beta, \quad \beta_1 \beta_2 = 0, \tag{11}$$

which we assume henceforth. We will further assume that β is not too close to 1, i.e.

$$\beta \le 1 - \epsilon, \quad \text{for some constant } \epsilon > 0. \tag{12}$$

The difference between the true gradient vector and its approximation is designated *gradient noise* and is denoted by:

$$\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1}) \overset{\Delta}{=} \nabla_w Q(\boldsymbol{\psi}_{i-1}; \boldsymbol{\theta}_i) - \nabla_w \mathbb{E}[Q(\boldsymbol{\psi}_{i-1}; \boldsymbol{\theta}_i)]. \tag{13}$$

Let the symbol $\boldsymbol{\mathcal{F}}_{i-1}$ represent the filtration generated by the random process $\boldsymbol{w}_j$ for $j \le i-1$:

$$\boldsymbol{\mathcal{F}}_{i-1} \overset{\Delta}{=} \text{filtration}\{\boldsymbol{w}_{-2}, \boldsymbol{\psi}_{-2}, \dots, \boldsymbol{w}_{i-1}, \boldsymbol{\psi}_{i-1}\}.$$

The following conditions on the gradient noise process essentially amount to requiring the approximation for the true gradient to be unbiased and for the variance of the gradient noise to decrease as the quality of the iterate improves. Both requirements are reasonable and they can be shown to be automatically satisfied in important cases, such as mean-square-error or logistic regression designs.

Assumption 2 (Conditions on gradient noise). *It is assumed that the first and second-order conditional moments of the gradient noise process satisfy the following conditions for any variable $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:*

$$\mathbb{E}[\boldsymbol{s}_i(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0 \tag{14}$$

$$\mathbb{E}[\|\boldsymbol{s}_i(\boldsymbol{w})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \le \gamma^2\|w^o - \boldsymbol{w}\|^2 + \sigma_s^2 \tag{15}$$

almost surely, for some nonnegative γ^2 and σ_s^2. ■

A useful step in the analysis that follows is to show first that recursions (7)-(8) can be transformed into a first-order recursion through a suitable change of variables. For this purpose, we first introduce the transformation matrices:

$$V = \begin{bmatrix} I_M & -\frac{\beta}{1-\beta} I_M \\ I_M & -\frac{1}{1-\beta} I_M \end{bmatrix},\ V^{-1} = \begin{bmatrix} \frac{1}{1-\beta} I_M & -\frac{\beta}{1-\beta} I_M \\ I_M & -I_M \end{bmatrix}.$$

Let $\widetilde{\boldsymbol{w}}_i = w^o - \boldsymbol{w}_i$ and define the transformed error vectors

$$\begin{bmatrix} \widehat{\boldsymbol{w}}_i \\ \check{\boldsymbol{w}}_i \end{bmatrix} \overset{\Delta}{=} V^{-1} \begin{bmatrix} \widetilde{\boldsymbol{w}}_i \\ \widetilde{\boldsymbol{w}}_{i-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{1-\beta}(\widetilde{\boldsymbol{w}}_i - \beta \widetilde{\boldsymbol{w}}_{i-1}) \\ \widetilde{\boldsymbol{w}}_i - \widetilde{\boldsymbol{w}}_{i-1} \end{bmatrix} \tag{16}$$

as well as:

$$\beta' \triangleq \beta\beta_1 + \beta - \beta_1, \tag{17}$$

$$\boldsymbol{H}_{i-1} \triangleq \int_0^1 \nabla_w^2 J(w^o - t\widetilde{\boldsymbol{\psi}}_{i-1})dt, \tag{18}$$

Using the mean-value theorem [6, 8], it can then be verified that recursions (7)–(8) can be transformed into the following extended recursion:

$$\begin{bmatrix}\widehat{\boldsymbol{w}}_i\\ \check{\boldsymbol{w}}_i\end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu_m}{1-\beta}\boldsymbol{H}_{i-1} & \frac{\mu_m\beta'}{(1-\beta)^2}\boldsymbol{H}_{i-1}\\ -\mu_m\boldsymbol{H}_{i-1} & \beta I_M + \frac{\mu_m\beta'}{1-\beta}\boldsymbol{H}_{i-1}\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{w}}_{i-1}\\ \check{\boldsymbol{w}}_{i-1}\end{bmatrix} + \mu_m\begin{bmatrix}\frac{\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1})}{1-\beta}\\ \boldsymbol{s}_i(\boldsymbol{\psi}_{i-1})\end{bmatrix} \tag{19}$$

The proof of the next result is omitted for brevity.

Theorem 1 (**Mean-square stability).** *Let Assumptions 1 and 2 hold and recall conditions (11) and (12). Then, for sufficiently small step-sizes μ_m, the momentum recursion* (7)-(8) *will converge exponentially to a small neighborhood of w^o:*

$$\limsup_{i\to\infty} \mathbb{E}\|w^o - \boldsymbol{w}_i\|^2 = O(\mu_m). \tag{20}$$

■

3. EQUIVALENCE IN THE QUADRATIC CASE

Theorem 1 establishes the convergence of the momentum recursions (7)-(8). But some important questions remain. Does the momentum implementation converge faster than the standard stochastic gradient method (2)? Does the momentum implementation lead to superior steady-state mean-square-deviation (MSD), measured in terms of the limiting value of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$? In this section we first examine the special case when $J(w)$ is quadratic to illustrate the main conclusions that will follow. We consider risks of the form:

$$J(w) = \frac{1}{2}\mathbb{E}\left(\boldsymbol{d}(i) - \boldsymbol{u}_i^\mathsf{T} w\right)^2, \tag{21}$$

where $\boldsymbol{d}(i)$ denotes a streaming sequence of zero-mean random variables with variance $\sigma_d^2 = \mathbb{E}\boldsymbol{d}^2(i)$, and $\boldsymbol{u}_i \in \mathbb{R}^M$ denotes a streaming sequence of independent zero-mean random vectors with covariance matrix $R_u = \mathbb{E}\boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T} > 0$. The cross covariance vector between $\boldsymbol{d}(i)$ and $\boldsymbol{u}_i$ is denoted by $r_{du} = \mathbb{E}\boldsymbol{d}(i)\boldsymbol{u}_i$. The data $\{\boldsymbol{d}(i), \boldsymbol{u}_i\}$ are assumed to be related via a linear regression model of the form:

$$\boldsymbol{d}(i) = \boldsymbol{u}_i^\mathsf{T} w^o + \boldsymbol{v}(i), \tag{22}$$

for some unknown w^o, and where $\boldsymbol{v}(i)$ is a zero-mean white noise process with power $\sigma_v^2 = \mathbb{E}\boldsymbol{v}^2(i)$ and assumed independent of $\boldsymbol{u}_j$ for all i, j.

In order to distinguish the variables for LMS from the variables for the momentum LMS version, which is described below in (26), we replace the notation $\boldsymbol{w}_i$ by $\boldsymbol{x}_i$. Then, the LMS recursion to solve (21) is given by

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \mu\boldsymbol{u}_i(\boldsymbol{d}(i) - \boldsymbol{u}_i^\mathsf{T}\boldsymbol{x}_{i-1}), \tag{23}$$

and the corresponding gradient noise is [8, 10]:

$$\boldsymbol{s}_i(\boldsymbol{x}) = (R_u - \boldsymbol{u}_i\boldsymbol{u}_i^\mathsf{T})(w^o - \boldsymbol{x}) - \boldsymbol{u}_i\boldsymbol{v}(i). \tag{24}$$

Subtracting w^o from both sides of (23), and setting $\widetilde{\boldsymbol{x}}_i = w^o - \boldsymbol{x}_i$, we obtain the error recursion:

$$\widetilde{\boldsymbol{x}}_i = (I_M - \mu R_u)\widetilde{\boldsymbol{x}}_{i-1} + \mu\boldsymbol{s}_i(\boldsymbol{x}_{i-1}). \tag{25}$$

On the other hand, if we apply the momentum recursions (7)–(8) to solve (21), we obtain from (16):

$$\begin{bmatrix}\widehat{\boldsymbol{w}}_i\\ \check{\boldsymbol{w}}_i\end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu_m}{1-\beta}R_u & \frac{\mu_m\beta'}{(1-\beta)^2}R_u\\ -\mu_m R_u & \beta I_M + \frac{\mu_m\beta'}{1-\beta}R_u\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{w}}_{i-1}\\ \check{\boldsymbol{w}}_{i-1}\end{bmatrix} + \mu_m\begin{bmatrix}\frac{\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1})}{1-\beta}\\ \boldsymbol{s}_i(\boldsymbol{\psi}_{i-1})\end{bmatrix} \tag{26}$$

Since we are assuming that the step-sizes $\{\mu, \mu_m\}$ and the momentum parameter β satisfy condition (3), then the first row of recursion (26) becomes:

$$\widehat{\boldsymbol{w}}_i = (I_M - \mu R_u)\widehat{\boldsymbol{w}}_{i-1} + \frac{\mu\beta'}{1-\beta}R_u\check{\boldsymbol{w}}_{i-1} + \mu\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1}). \tag{27}$$

Comparing with the LMS recursion (25), we find that both relations are quite similar, except that the momentum recursion has an extra driving term dependent on $\check{\boldsymbol{w}}_{i-1}$. However, recall from (16) that $\check{\boldsymbol{w}}_{i-1} = \widetilde{\boldsymbol{w}}_{i-1} - \widetilde{\boldsymbol{w}}_{i-2}$, which is the difference between two consecutive points generated by momentum LMS. Intuitively, it is not hard to see that $\check{\boldsymbol{w}}_{i-1}$ is in the order of $O(\mu)$, which makes $\mu R_u\check{\boldsymbol{w}}_{i-1}$ in the order of $O(\mu^2)$. When the step-size μ is small, this $O(\mu^2)$ term can be ignored. Consequently, the above recursions for $\widehat{\boldsymbol{w}}_i$ and $\widetilde{\boldsymbol{x}}_i$ should evolve close to each other. This observation can be established more formally as follows; again, we omit the proof for brevity.

Theorem 2 (**Equivalence for quadratic costs).** *Consider the standard and momentum stochastic gradient methods to solve problem* (21). *Assume the algorithms start from the same initial states, namely, $\boldsymbol{\psi}_{-2} = \boldsymbol{w}_{-2} = \boldsymbol{x}_{-1}$. Suppose conditions (11) and (12) hold, and that the step-sizes $\{\mu, \mu_m\}$ satisfy (3). Then, it holds for sufficiently small μ that*

$$\mathbb{E}\|\boldsymbol{w}_i - \boldsymbol{x}_i\|^2 = O(\mu^2), \quad \forall i = 0, 1, 2, 3, \ldots \tag{28}$$

■

4. EQUIVALENCE IN THE GENERAL CASE

We now extend the analysis to more general risks. The analysis in this case is more demanding because the Hessian matrix of $J(w)$ is now $w-$dependent. Nevertheless, under some smoothness conditions, it is still possible to establish a strong equivalence result. The first condition below replaces Assumption 2; actually, it can be verified that if Assumption 3 holds, then Assumption 2 will also hold.

Assumption 3 (**Conditions on gradient noise).** *It is assumed that the first and fourth-order conditional moments of the gradient noise process satisfy the following conditions for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$:*

$$\mathbb{E}[\boldsymbol{s}_i(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0 \tag{29}$$

$$\mathbb{E}[\|\boldsymbol{s}_i(\boldsymbol{w})\|^4|\boldsymbol{\mathcal{F}}_{i-1}] \leq \gamma_4^4\|w^o - \boldsymbol{w}\|^4 + \sigma_{s,4}^4 \tag{30}$$

almost surely, for some nonnegative constants γ_4^4 and $\sigma_{s,4}^4$.

■

The next two assumptions introduce smoothness conditions on the gradient noise process and the Hessian matrix of the risk function.

Assumption 4. *Consider the iterate $\boldsymbol{\psi}_{i-1}$ that is generated by the momentum recursion* (7)*. Let $\boldsymbol{x}_{i-1}$ denote the iterate that is generated by the gradient recursion* (2) *(we are writing $\boldsymbol{x}_i$ instead of $\boldsymbol{w}_i$). It is assumed that the gradient noise process satisfies:*

$$\mathbb{E}\|\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1}) - \boldsymbol{s}_i(\boldsymbol{x}_{i-1})\|^2 \leq \xi_1 \mathbb{E}\|\boldsymbol{\psi}_{i-1} - \boldsymbol{x}_{i-1}\|^2, \tag{31}$$

$$\mathbb{E}\|\boldsymbol{s}_i(\boldsymbol{\psi}_{i-1}) - \boldsymbol{s}_i(\boldsymbol{x}_{i-1})\|^4 \leq \xi_2 \mathbb{E}\|\boldsymbol{\psi}_{i-1} - \boldsymbol{x}_{i-1}\|^4. \tag{32}$$

for some nonnegative constants ξ_1 and ξ_2. ■

Assumption 5. *The Hessian of the risk function $J(w)$ in* (1) *is Lipschitz continuous, i.e.*

$$\|\nabla_w^2 J(w_1) - \nabla_w^2 J(w_2)\| \leq \kappa \|w_1 - w_2\|. \tag{33}$$

for some constant $\kappa \geq 0$. ■

The above three assumptions hold automatically for important cases, such as least-mean-squares and logistic regression problems. The following main result can now be established; the proof is omitted.

Theorem 3 (Equivalence for general risks). *Consider the standard and momentum stochastic gradient recursions* (2) *and* (7)–(8) *and assume they start from the same initial states, namely, $\boldsymbol{\psi}_{-2} = \boldsymbol{w}_{-2} = \boldsymbol{x}_{-1}$. Suppose conditions (11), (12), and (3) hold. Under Assumptions 1, 3, 4, and 5, and for sufficiently small step-sizes, it holds that*

$$\mathbb{E}\|\boldsymbol{w}_i - \boldsymbol{x}_i\|^2 = O(\mu^{3/2}), \quad \forall i = 0, 1, 2, 3, \ldots \tag{34}$$

Furthermore, in the limit,

$$\limsup_{i\to\infty} \mathbb{E}\|\boldsymbol{w}_i - \boldsymbol{x}_i\|^2 = O(\mu^2). \tag{35}$$

■

We remarked earlier in Subsection 1.1 that with the same step-size μ, the momentum method converges faster than the traditional stochastic-gradient method but attains worse mean-square-deviation (MSD) performance in steady-state. This behavior is consistent with the result of Theorem 3, which allows us to interpret the momentum implementation as corresponding to a stochastic-gradient implementation with the larger step-size $\mu/(1-\beta)$. There is also a second more intuitive explanation as to why the momentum variant leads to worse steady-state performance. While the momentum terms $\boldsymbol{w}_i - \boldsymbol{w}_{i-1}$ and $\boldsymbol{\psi}_i - \boldsymbol{\psi}_{i-1}$ in (7)–(8) help smooth the convergence trajectories, and hence accelerate the convergence rate, they nevertheless introduce additional noise into the evolution of the algorithm because all iterates $\boldsymbol{w}_i$ and $\boldsymbol{\psi}_i$ are distorted by perturbations.

One should notice that Theorem 3 is affirming that when a sufficiently small step-size μ is employed, the iterates of the standard and momentum stochastic gradient methods will evolve close to each other. In this sense, both algorithms are regarded as equivalent. However, when μ is relatively large, the gap between both algorithms need not be negligible. In fact, previous literature [18, 19] observed that momentum LMS with relatively large step-size, compared to standard LMS, can have a smoothing effect when the Hessian of the cost function has a bad condition number or the measurement noise includes abrupt impulses.

5. EXPERIMENTAL RESULTS

To avoid deteriorating the MSD performance while retaining the acceleration advantages of momentum, the results so far suggest employing a *decaying* momentum factor. In this section we illustrate this and other conclusions by considering a regularized logistic regression problem. In this example, we are interested in minimizing

$$J(w) \triangleq \frac{\rho}{2}\|w\|^2 + \mathbb{E}\Big\{ \ln\big[1 + \exp(-\boldsymbol{\gamma}(i)\boldsymbol{h}_i^{\mathsf{T}} w)\big] \Big\} \tag{36}$$

where the approximate gradient is given in Equation (2.11) in [8].

In the simulation, we generate 20000 samples $(\boldsymbol{h}_i, \boldsymbol{\gamma}(i))$. Among these training points, 10000 feature vectors $\boldsymbol{h}_i$ correspond to label $\boldsymbol{\gamma}(i) = 1$ and each $\boldsymbol{h}_i \sim \mathcal{N}(1.5 \times \mathbb{1}_{10}, R_h)$ for some diagonal covariance R_h. The remaining 10000 feature vectors $\boldsymbol{h}_i$ correspond to label $\boldsymbol{\gamma}(i) = -1$ and each $\boldsymbol{h}_i \sim \mathcal{N}(-1.5 \times \mathbb{1}_{10}, R_h)$. Besides, we set $\rho = 0.01$. The optimal solution w^o is computed via the traditional gradient descent method. All simulation results shown below are averaged over 100 trials.

We first compare the standard and momentum stochastic methods using $\mu = \mu_m = 0.01$. The momentum parameter β is set to 0.5. These two methods are illustrated in Fig. 1 with blue and red curves, respectively. It is seen that the momentum method converges faster, but the MSD performance is much worse. Next, we set $\mu_m = \mu(1-\beta) = 0.005$ and illustrate this case with the magenta curve. It is observed that the magenta and blue curves are indistinguishable, which confirms the equivalence predicted by Theorem 3. Finally we illustrate an implementation with a decaying momentum parameter $\beta(i)$ by the green curve. In this simulation, we set $\mu_m = 0.01$ and make $\beta(i)$ decrease in a stair-wise manner: when $i \in [1, 500]$, $\beta(i) = 0.5$; when $i \in [501, 1000]$, $\beta(i) = 0.5/(501^{0.3})$; when $i \in [1001, 1500]$, $\beta(i) = 0.5/(1001^{0.3})$; ...; when $i \in [9501, 10000]$, $\beta(i) = 0.5/(9501^{0.3})$. With this decaying $\beta(i)$, it is seen that the momentum method recovers its faster convergence rate and attains the same steady-state MSD performance as the stochastic-gradient implementation.

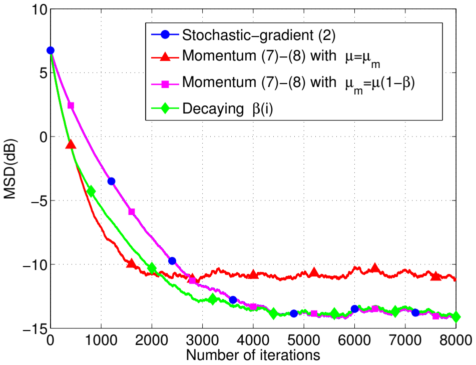

Fig. 1. Convergence behavior of standard and momentum stochastic gradient methods.

6. CONCLUDING REMARKS

The results in this work establish that momentum methods are equivalent to the standard stochastic gradient method at *all* time instants, namely, their trajectories evolve close to each other for sufficiently small step-sizes. This conclusion holds for general risk functions, and is not limited to quadratic risks. The analysis further suggests a method to enhance performance in the stochastic setting by tuning the momentum parameter over time.

7. REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, NJ, 1985.

[2] S. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, NJ, Fourth Edition, 2008.

[3] A. H. Sayed, *Adaptive Filters*, Wiley, NY, 2008.

[4] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, Academic Press, NY, 2015.

[5] Y. Z. Tsypkin and Z. J. Nikolic, *Adaptation and Learning in Automatic Systems*, Academic Press, NY, 1971.

[6] B. T. Polyak, *Introduction to Optimization*, Optimization Software, NY, 1987.

[7] D. P. Bertsekas, "Nonlinear programming," 1999.

[8] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[9] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," in *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, 2008, pp. 161–168.

[10] A. H Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[11] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[12] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[13] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2004.

[14] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[15] J. G. Proakis, "Channel identification for high speed digital communications," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 916–922, 1974.

[16] R. Sharma, W. A. Sethares, and J. A. Bucklew, "Analysis of momentum adaptive filtering algorithms," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1430–1434, 1998.

[17] J. J. Shynk and S. Roy, "The LMS algorithm with momentum updating," in *Proc. IEEE International Symposium on Circuits and Systems*, Espoo, Finland, June 1988, pp. 2651–2654.

[18] S. Roy and J. J. Shynk, "Analysis of the momentum LMS algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 12, pp. 2088–2098, 1990.

[19] M. A. Tugay and Y. Tanik, "Properties of the momentum LMS algorithm," *Signal Processing*, vol. 18, no. 2, pp. 117–127, 1989.

[20] M. Bellanger, *Adaptive Digital Filters and Signal Analysis*, 2nd Edition, Marcel Dekker, 2001.

[21] W. Wiegerinck, A. Komoda, and T. Heskes, "Stochastic dynamics of learning with momentum in neural networks," *Journal of Physics A: Mathematical and General*, vol. 27, no. 13, pp. 4425–4438, 1994.

[22] L. K. Ting, C. F. N. Cowan, and R. F. Woods, "Tracking performance of momentum LMS algorithm for a chirped sinusoidal signal," in *Proc. European Signal Processing Conference*, Tampere, Finland, 2000, pp. 1–4.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. International Conference on Machine Learning*, Atlanta, USA, 2013, pp. 1139–1147.