Stochastic optimization under time drift: iterate averaging, step decay, and high probability guarantees

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of minimizing a convex function that is evolving in time according to unknown and (possibly) stochastic dynamics. Such problems abound in the machine learning and signal processing literature, under the names of adaptive filtering, stochastic tracking, and concept drift. We provide novel non-asymptotic convergence guarantees for stochastic algorithms with iterate averaging. We prove both bounds in expectation and in high probability. The latter hold under light tail assumptions and without any restriction on the size of the domain. We characterize the interplay of the learning rate, the gradient noise, and the rate of variation over time. Numerical examples illustrate the theoretical results.

1 Introduction

Stochastic optimization underpins much of machine learning theory and practice, in large part due to its flexibility at tackling a wide array of learning tasks. Significant progress has been made over the last two decades in the finite-time analysis of stochastic approximation algorithms [6, 8, 7, 33, 1, 24]. [2, 10, 25]. The predominant assumption in stochastic optimization for machine learning is that the distribution generating the data is fixed throughout the run of the process [6, 2, 32].. There is no shortage of problems, however, where this assumption is grossly violated for reasons beyond the learner's control. Indeed, data often shifts and evolves over time for reasons that are independent of the learning process.

Two examples are worth highlighting. The first is a classical problem in signal processing—called adaptive filtering or stochastic tracking [30, 22]—wherein the learning algorithm aims to track over time a moving target driven by an unknown stochastic process. The second example is the concept drift phenomenon [31, 16] in online learning, wherein the true hypothesis may be changing over time, as in topic modeling or spam classification. An important goal in online problems, and the one we adopt here, is to track as closely as possible an unknown sequence of minimizers or minimal values. The tracking error efficiency of stochastic algorithms in online settings is much less developed than sample complexity guarantees for static problems.

We present finite-time efficiency estimates in expectation and in high-probability for the tracking error of the proximal stochastic gradient method under time drift. Our results concisely explain the interplay between the learning rate, the noise variance in the gradient oracle, and the strength of the time drift. The high probability results merely assume that the gradient noise and distribution drift have light tails. Moreover, none of the results require the objectives to have bounded domains, thereby allowing to incorporate non-smooth regularizers in the problem formulations. While conventional wisdom and previous work recommend the use of constant step-sizes under time drift, we show that in important regimes the best step size schedule in hindsight is one that is exponentially decaying to an "optimal step size".

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Related work.

Our current work fits within the broader literature on stochastic tracking, online optimization with controlled increments, and high probability guarantees in stochastic optimization. We now survey the most relevant literature in these areas.

Stochastic tracking and adaptive filtering Stochastic gradient type algorithms for stochastic tracking and adaptive filtering have been the subject of extensive research in the past century, leading to major technological advances in digital signal processing and global wireless communications. Most works have focused on the so-called least mean-squares (LMS) algorithm and its variants, which can be viewed as a stochastic gradient method on a least-squares loss based objective. Other stochastic algorithms that have been studied in these setting with a larger cost per iteration include recursive least-squares and the Kalman filter [13]. Recent works have revisited these methods from a more modern viewpoint [35, 27]. In contrast to these works, we (1) work with a wider class of objectives that encompass constrained or regularized empirical risk minimization with a proximal-friendly regularizer, (2) make only mild assumptions on the gradient noise and do not require it to be bounded, (3) impose transparent conditions on the rate of variation of the minimizers themselves, and (4) do not require the method to sample from a stale distribution.

Online optimization with controlled increments Online learning under concept drift was first considered by [26] and further developed in several papers [19, 4]. In this literature, the data distribution is fixed over time, and the rate of variation is stated in terms of the probability of disagreement of consecutive target functions which is assumed to be upper bounded. Another, different, line of work, assume a time partitioning, an expert in each time interval, and the goal is to compete with the expert in each segment. Closer to this work is [16, 18] whose framework is online convex optimization and state bounds in terms of maximum regret over any contiguous time interval; see also [9, 29, 5]. In contrast to these works, in our framework, we state our bounds in the same spirit as in classical stochastic approximation, that is in terms of distance to optimum and objective function gap, and present results both in terms of expectation and in high-probability.

High probability guarantees in stochastic optimization A large part of our work revolves around high probability guarantees for stochastic optimization. Classical references on the subject in static settings and for minimizing regret in online minimization include [23, 28, 3, 17]. When the domain of the objective function is bounded, high probability guarantees typically follow directly from the Azuma–Hoeffding inequality. In the unbounded case, a variety of alternative techniques exist based on Freedman's inequality and doubling tricks; see e.g. [3, 17]. A more recent line of work [14] establishes a generalized Freedman inequality that is custom tailored for analyzing stochastic gradient-type methods, and results in best known high probability guarantees for problems with unbounded domains. Our arguments closely follow the paradigm of [14] based on the generalized Freedman inequality.

2 Framework and assumptions

Stochastic optimization under time drift. Throughout the paper, we consider the sequence of stochastic optimization problems

$$\min_x \ \varphi_t(x) := f_t(x) + r_t(x), \tag{1}$$

where the time index runs over the horizon $t \in \mathbb{N}$. In typical circumstances arising in machine learning and signal processing, the function $f_t \colon \mathbb{R}^d \to \mathbb{R}$ corresponds to an average loss that varies in time, while $r_t \colon \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ models constraints or promotes structure (e.g. sparsity) in the variable x. Throughout, we make the standard assumption that $r_t \colon \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ are convex, $f_t \colon \mathbb{R}^d \to \mathbb{R}$ are μ-strongly convex, and that the gradient ∇f_t is L-Lipschitz continuous. The minimizer and minimal value of (1) will be denoted by $x_t^\star$ and $\varphi_t^\star$, respectively. As motivation, we describe two classical examples of (1) that are worth keeping in mind and that guide our framework: stochastic tracking of a drifting target and online learning under concept drift.

Example 2.1 (Stochastic tracking of a drifting target)**.** The problem of stochastic tracking, also called adaptive filtering in the signal processing, is to track a moving target $x_t^\star$ from observations

$$b_t = c_t(x_t^\star) + \epsilon_t,$$

Algorithm 1 Online Proximal SG $SG(X_0, \{\eta_t\}, T))$

Input: initial x_0 and step-size sequence $\{\eta_t\}_{t=0}^T \subset (0, \infty)$.
Step $t = 0, \ldots, T$:

$$\text{Sample } g_t = \widetilde{\nabla} f_t(x_t)$$
$$\text{Set } x_{t+1} = \text{prox}_{\eta_t r_t}(x_t - \eta_t g_t)$$

where $c_t(\cdot)$ is a known measurement map and ϵ_t is a mean zero noise vector. A typical time dependent problem formulation takes the form

$$\min_x \ \mathop{\mathbb{E}}_{\epsilon_t} \ell(b_t - c_t(x)) + r_t(x),$$

where ℓ is the negative log-likelihood function of ϵ_t and $r_t(\cdot)$ encodes available side information about the target $x_t^\star$. Most common choices for r_t are the ℓ_1-norm and the squared ℓ_2-norm. The motion of the target $x_t^\star$ is typically driven by a random walk or a diffusion [13, 30].

Example 2.2 (Online learning under distribution drift)**.** The problem of online learning under a distributional drift is to learn while the data distribution may change over time. More formally, one problem formulation takes the form

$$\min_x \ \mathop{\mathbb{E}}_{z \sim D(w_t)} \ell(x, z) + r_t(x).$$

where $D(w_t)$ is a data distribution that depends on an unknown parameter sequence (w_t), which itself may evolve stochastically. The drift is often assumed to be piecewise constant in online learning [31, 16].

The main goal of a learning algorithm for problem (1) is to generate a sequence of points x_t in order to minimize some natural performance metric. The most prevalent performance metrics in the literature are the *tracking error* and the *dynamic regret*. We will primarily focus on two types of tracking error, $\|x_t - x_t^\star\|^2$ and $\varphi_t(x_t) - \varphi_t(x_t^\star)$. Our standing assumption is that at every time t, and for any query point x, the learner may obtain an unbiased estimator $\widetilde{\nabla} f_t(x) \in \mathbf{R}^d$ of the true gradient $\nabla f_t(x)$, in order to proceed with a stochastic gradient like optimization algorithm. With this oracle access, the online stochastic gradient algorithm (Algorithm 1) in each iteration t simply takes a stochastic gradient step on f_t at x_t followed by a proximal operation on r_t. The goal of our work is to obtain efficiency estimates for this procedure that hold both in expectation and with high probability.

The guarantees we obtain allow both the iterates x_t *and the minimizers* $x_t^\star$ to evolve stochastically. This is convenient for example when tracking a moving target $x_t^\star$, whose motion may be governed by a stochastic process such as a random walk or a diffusion (see Example 2.1).

2.1 Running assumption on the stochastic process.

Setting the stage, we let $z_t := \nabla f_t(x_t) - g_t$ denote the gradient noise and we impose the following assumption modeling stochasticity in the online problem.

Assumption 2.3. There exists a filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ with filtration $\mathbb{F} = (\mathcal{F}_t)_{t \geq 0}$ such that $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and the following holds for all $t \geq 0$:

(i) $x_t, x_t^\star, \nabla f_t(x_t) : \Omega \to \mathbf{R}^d$ are $\mathcal{F}_t$-measurable.

(ii) $z_t : \Omega \to \mathbf{R}^d$ is $\mathcal{F}_{t+1}$-measurable with $\mathbb{E}[z_t \,|\mathcal{F}_t] = 0$.

Assumption (i) simply says that $x_t, x_t^\star, \nabla f_t(x_t)$ are fully determined by information up to time t. Assumption (ii) asserts that the noise z_t is fully determined by information up to time $t+1$ and has zero mean conditioned on the information up to time t. For example, this assumption holds automatically in Example 2.2.

Clearly, the efficiency estimates of Algorithm 1 must take into account the variations of the function f_t in time t. The two most popular metrics for measuring time variations is the distance between successive minimizers, $\|x_t^\star - x_{t+1}^\star\|$, and gradient variations in time, $\sup_x \|\nabla f_{t+1}(x) - \nabla f_t(x)\|$. An estimate on the gradient variation always implies an estimate on the distance between minimizers.

Observation 2.4 (Gradient variation vs. minimizer variation)**.** Suppose that the regularizers $r_t \equiv r$ are identical for all times $t \in \mathbb{N}$. Then for any indices i and t, the estimate holds:

$$\mu\|x_i^\star - x_t^\star\| \leq \|\nabla f_i(x_t^\star) - \nabla f_t(x_t^\star)\|.$$

2.2 A simple bound on the tracking error.

To summarize our ideas, and set the groundwork for high probability guarantees, we first state here a simple finite-time bound on the expected tracking error $\mathbb{E}\|x_t - x_t^\star\|^2$. For simplicity, we assume that the moments, $\mathbb{E}\|z_t\|^2$ and $\mathbb{E}\|x_t^\star - x_{t+1}^\star\|^2$, are uniformly bounded and the learning rate $\eta = \eta_t$ is constant. More general guarantees that take into account weighted averages of the moments and allow for time dependent learning rates can be found in the Supplement.

Assumption 2.5 (Bounded second moment)**.** There exist finite $\sigma, \Delta \geq 0$ satisfying $\mathbb{E}\|z_t\|^2 \leq \sigma^2$ and $\mathbb{E}\|x_t^\star - x_{t+1}^\star\|^2 \leq \Delta^2$ for all $t \geq 0$.

The following theorem establishes a one-step improvement guarantee for Algorithm 1, and serves as the basis for much of what follows.

Theorem 2.6 (Expected distance)**.** *Suppose that Assumption 2.5 holds. Then the iterates produced by Algorithm 1 with constant learning rate $\eta \leq \frac{1}{2L}$ satisfy the bound:*

$$\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \underbrace{(1-\mu\eta)^t\|x_0 - x_0^\star\|^2}_{\textit{optimization}} + \underbrace{\frac{\eta\sigma^2}{\mu}}_{\textit{noise}} + \underbrace{\left(\frac{\Delta}{\mu\eta}\right)^2}_{\textit{drift}}.$$

Interplay of optimization, noise, and drift. Theorem 2.6 states that when using a constant learning rate, the error $\mathbb{E}\|x_t - x_t^\star\|^2$ decays linearly in time t, until it reaches the "noise+drift" error $\eta\sigma^2/\mu + (\Delta/\mu\eta)^2$. Notice that the "noise+drift" error can not be be made arbitrarily small. This is perfectly in line with intuition: a learning rate that is too small prevents the algorithm from catching up with the minimizers $x_t^\star$. We note that each of the error terms due to the optimization, noise, and drift are known to be tight for the stochastic gradient method.

With Theorem 2.6 at hand, we may define the optimal asymptotic tracking error of Algorithm 1 as

$$\mathcal{E} := \min_{\eta \in (0, 1/2L]} \left\{ \frac{\eta\sigma^2}{\mu} + \left(\frac{\Delta}{\mu\eta}\right)^2 \right\}. \tag{2}$$

A short computation shows that the corresponding asymptotically optimal choice of η is given by $\eta_\star = \min\{1/2L, (2\Delta^2/\mu\sigma^2)^{1/3}\}$. Plugging $\eta_\star$ into the definition of $\mathcal{E}$, we see that algorithm enjoys qualitatively different behaviors in settings corresponding to low or high drift-to-noise ratio Δ/σ, explicitly given by

$$\mathcal{E} \asymp \begin{cases} \frac{\sigma^2}{\mu L} + \left(\frac{L\Delta}{\mu}\right)^2 & \text{if } \frac{\Delta}{\sigma} \geq \sqrt{\frac{\mu}{16L^3}} \\ \left(\frac{\Delta\sigma^2}{\mu^2}\right)^{2/3} & \text{otherwise.} \end{cases}$$

Two regimes of variation are brought to light by the above results: a high drift-to-noise regime, and a low drift-to-noise regime. The high drift-to-noise setting is uninteresting from the viewpoint of stochastic optimization because the optimal choice of stepsize is as large as in the deterministic setting $\eta_\star = 1/2L$. In contrast, the low drift-to-noise regime is interesting because it necessitates using a learning rate $\eta_\star = (2\Delta^2/\mu\sigma^2)^{1/3}$ that is smaller than $1/2L$ and exhibits a nontrivial scaling with the problem parameters.

Learning rates vs. rate of variation. A central question is to find a step size or learning rate schedule that achieves a tracking error $\mathbb{E}\|x_t - x_t^\star\|^2$ that is within a constant factor of $\mathcal{E}$ in the shortest possible time. The answer is clear in the high drift-to-noise setting $\frac{\Delta}{\sigma} \geq \sqrt{\frac{\mu}{16L^3}}$. Indeed, in this case, Algorithm 1 with a constant learning rate $\eta = 1/2L$ will find a point x_t satisfying $\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \mathcal{E}$ in time $t \lesssim \frac{L}{\mu}\log(\|x_0 - x_0^\star\|^2/\mathcal{E})$. Notice that this rate is linear in the tracking accuracy $\mathcal{E}$; intuitively, the reason for the absence of a sublinear component is that the error due to the drift Δ dominates the error due to the variance σ^2 in the stochastic gradient.

The low drift-to-noise setting $\Delta/\sigma < \sqrt{\mu/16L^3}$ is more subtle. Namely, the simplest strategy is to execute Algorithm 1 with the constant learning rate $\eta_\star = (2\Delta^2/\mu\sigma^2)^{1/3}$. Then a direct application of Theorem 2.6 yields the estimate $\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \mathcal{E}$ after time $t \lesssim \frac{\sigma^2}{\mu^2\mathcal{E}} \log(\|x_0 - x_0^\star\|^2/\mathcal{E})$. This efficiency estimate can be significantly improved by gradually decaying the learning rate using a "step-decay schedule", wherein the algorithm is implemented in epochs with the new learning rate chosen to be the midpoint between the current learning rate and $\eta_\star$. Such schedules are well known to improve efficiency in the static setting, as was discovered in [11, 12], and can be used here. The end result is the following theorem, whose formal statement and proof appear in the Supplement.

Theorem 2.7 (Informal)**.** *Suppose that Assumption 2.5 holds. Then there is a learning rate schedule $\{\eta_t\}$ such that Algorithm 1 produces a point x_t satisfying $\mathbb{E}\|x_t - x_t^\star\|^2 \lesssim \mathcal{E}$ after time $t \lesssim \frac{L}{\mu} \log\left(\frac{\|x_0 - x_0^\star\|^2}{\mathcal{E}}\right) + \frac{\sigma^2}{\mu^2\mathcal{E}}$.*

Remarkably, the efficiency estimate in Theorem 2.7 looks identical to the efficiency estimate in the classical static setting [23], with $\mathcal{E}$ playing the role of the target accuracy ε. Theorems 2.6 and 2.7 provide useful baseline guarantees for the performance of Algorithm 1. Nonetheless, these guarantees are all stated in terms of the *expected* tracking error $\mathbb{E}\|x_t - x_t^\star\|^2$, and are therefore only meaningful if the entire algorithm can be repeated from scratch multiple times. There is no shortage of situations, when a learning algorithm is operating in real time and the time drift is irreversible; in such settings, the algorithm may only be executed once. Such settings call for efficiency estimates that hold with high probability, rather than in expectation. This is the content of the next section.

3 Finite-time guarantees in high probability

We present in this section high-probability guarantees on the tracking error. To this end, we make the following standard light tail assumptions on the gradient noise and the time drift [20, 21, 34].

Assumption 3.1 (Light-tails)**.** Assume that there exist constants $\Delta, \sigma > 0$ satisfying the following.

1. **(Drift)** The random variable $\Delta_t^2 = \|x_t^\star - x_{t+1}^\star\|^2$ is sub-exponential conditioned on $\mathcal{F}_t$ with parameter Δ^2, meaning
$$\mathbb{E}[\exp(\lambda\Delta_t^2)|\mathcal{F}_t] \le \exp(\lambda\Delta^2) \text{ for all } 0 \le \lambda \le \Delta^{-2}.$$
2. **(Noise)** The gradient noise z_t is norm sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter σ:
$$\mathbb{P}\{\|z_t\| \ge \tau|\mathcal{F}_t\} \le 2e^{-\frac{\tau^2}{2\sigma^2}} \quad \text{for all} \quad \tau > 0.$$

It is worthwhile to note some common settings in which Assumption 3.1 holds; the claims right below follow from classical results on sub-Gaussian r.v.-s [34]. Namely, fix constants $\sigma, \Delta > 0$. If Δ_t is bounded by Δ, then clearly Δ_t is sub-exponential with parameter Δ. Similarly if $\|z_t\|$ is bounded by σ, then z_t is norm sub-Gaussian with parameter σ. Alternatively, if the increment $x_t^\star - x_{t+1}^\star$ is mean-zero sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter $\Delta/\sqrt{d}$, then $x_t^\star - x_{t+1}^\star$ is mean-zero norm sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter $2\sqrt{2} \cdot \Delta$ and hence Δ_t^2 is sub-exponential conditioned on $\mathcal{F}_t$ with parameter $c \cdot \Delta^2$ for some absolute constant $c > 0$. Similarly, z_t is sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter $\frac{\sigma}{2\sqrt{2d}}$, then z_t is norm sub-Gaussian conditioned on $\mathcal{F}_t$ with parameter σ.

3.1 Tracking the minimizer

The theorem below shows that if Assumption 3.1 holds, then bounds on expectation $\mathbb{E}\|x_t - x_t^\star\|^2$, derived in Theorem 2.6, hold with high probability.

Theorem 3.2 (High probability distance tracking)**.** *Suppose that Assumption 3.1 holds and let $\{x_t\}$ be the iterates produced by Algorithm 1 with constant learning rate $\eta \le 1/2L$. Then there is a numerical constant c such that for any specified $t \in \mathbb{N}$ and for any $\delta \in (0, 1)$, the estimate*

$$\|x_t - x_t^\star\|^2 < c\left(\left(1 - \frac{\mu\eta}{2}\right)^t \|x_0 - x_0^\star\|^2 + \frac{2}{\mu}\left(\eta\sigma^2 + \frac{\Delta^2}{\mu\eta^2}\right)\log\left(\frac{e}{\delta}\right)\right),$$

holds with probability at least $1-\delta$.

We note that in the special case when the domains of r_t are uniformly bounded and the noise z_t is bounded, a sub-Gaussian tail bound on $\|x_t - x_t^\star\|^2$ follows from a direct application of the Azuma-Hoeffding inequality. Since we do not impose such boundedness assumptions, we use a different argument that closely follows that of [15]. The main idea is to build a careful recursion for the moment generating function of $\|x_t - x_t^\star\|^2$, leading to subexponential tail bound. As a consequence of Theorem 3.2, we can again consider a step-decay schedule to obtain the following efficiency estimate with high probability.

Theorem 3.3 (Informal)**.** *Suppose that Assumption 3.1 holds. Then there is a learning rate schedule* $\{\eta_t\}$ *such that for any specified* $t \in \mathbb{N}$ *and* $\delta \in (0,1)$*, Algorithm 1 produces a point* x_t *satisfying*

$$\|x_t - x_t^\star\|^2 \lesssim \mathcal{E}\log(Ke/\delta)$$

with probability at least $1-\delta$ *after a time*

$$t \lesssim \frac{L}{\mu}\log\left(\frac{\|x_0 - x_0^\star\|^2}{\mathcal{E}}\right) + \frac{\sigma^2}{\mu^2\mathcal{E}}, \quad \textit{where } K \lesssim \log_2\left(\frac{1}{L}\cdot\left(\frac{\sigma^2\mu}{\Delta^2}\right)^{1/3}\right)$$

3.2 Tracking the minimum

The results outlined so far have focused on tracking the minimizer $x_t^\star$. In this section, we present results for tracking the minimal value $\varphi_t^\star$. The two goals are fundamentally different in the stochastic setting. Generally speaking, good bounds on the function gap along with strong convexity imply good bounds on the distance to the minimizer; the reverse implication is false. To this end, we will require stronger assumptions on the drift. Most importantly, we will assume control on the gradient gap $\sup_x \|\nabla f_t(x) - \nabla f_{t+1}(x)\|$ rather than on the minimizer gap $\|x_t^\star - x_{t+1}^\star\|^2$. We work here with $\Delta := B/\mu$. The assumptions below are common in the online learning literature [16].

Assumption 3.4. Suppose that the regularizers $r_t \equiv r$ are identical for all times t and that there exists constants $B, \sigma > 0$ such that that the following properties hold for all indices $0 \le i < t$.

1. **(Drift)** The gradient variation $G_{i,t} := \sup_x \|\nabla f_t(x) - \nabla f_i(x)\|$ satisfies the bound $\mathbb{E}\left[G_{i,t}^2\right] \le B^2|t-i|^2$.
2. **(Noise)** The noise z_t satisfies $\mathbb{E}\|z_t\|^2 \le \sigma^2$ and is uncorrelated with the drift, meaning $\mathbb{E}\langle z_i, x_t^\star\rangle = 0$.

These two assumption are natural indeed. The assumption on the drift, intuitively, asserts that gradient deviation $G_{i,t}$ can grow only linearly in time $|t-i|$. In particular, returning to Example 2.2, suppose that the distribution map $D(\cdot)$ is γ-Lipschitz continuous in the Wasserstein-1 distance and the gradient $\nabla\ell(x,\cdot)$ is β-Lipschitz continuous for all x. Then the Kantorovich-Rubinstein Theorem directly implies $\mathbb{E}\|G_{i,t}\|^2 \le \gamma^2\beta^2\mathbb{E}\|w_i - w_t\|^2$. Therefore, as long as long as the variance $\mathbb{E}\|w_i - w_t\|^2$ scales quadratically in $|i-t|$, the desired assumption holds. The assumption on the noise requires $\mathbb{E}\|z_t\|^2$ to be finite and for the condition $\mathbb{E}\langle z_i, x_t^\star\rangle = 0$ to hold. The latter property holds automatically if the noise and the drift evolve independently of each other, as is the case for instance in Example 2.2. Taking into account Lemma 2.4, it is clear that Assumption 3.4 implies the earlier Assumption 2.5 with the same constants Δ, σ.

We now present a one-step improvement guarantee for tracking the optimal value along the running average $\hat{x}_t$ of the iterates x_t, defined as

$$\hat{x}_0 = x_0 \qquad \text{and} \qquad \hat{x}_t = \left(1 - \tfrac{\mu\eta}{2-\mu\eta}\right)\hat{x}_{t-1} + \tfrac{\mu\eta}{2-\mu\eta}x_t \quad \forall t \ge 1.$$

Theorem 3.5 (Expected function gap)**.** *Suppose that Assumption 3.4 holds. Let* $\{x_t\}$ *be the iterates produced by Algorithm 1 with constant learning rate* $\eta \in (0, 1/2L]$*. Then for all* $t \ge 0$*, the average iterate* $\hat{x}_t$ *satisfies:*

$$\mathbb{E}[\varphi_t(\hat{x}_t) - \varphi_t^\star] \lesssim \underbrace{\left(1 - \frac{\mu\eta}{2}\right)^t \cdot \mathbb{E}\left[\varphi_t(x_0) - \varphi_t^\star + \frac{\mu}{4}\|x_0 - x_t^\star\|^2\right]}_{\textit{optimization}} + \mu\left(\underbrace{\frac{\eta\sigma^2}{\mu}}_{\textit{noise}} + \underbrace{\left(\frac{\Delta}{\mu\eta}\right)^2}_{\textit{drift}}\right).$$

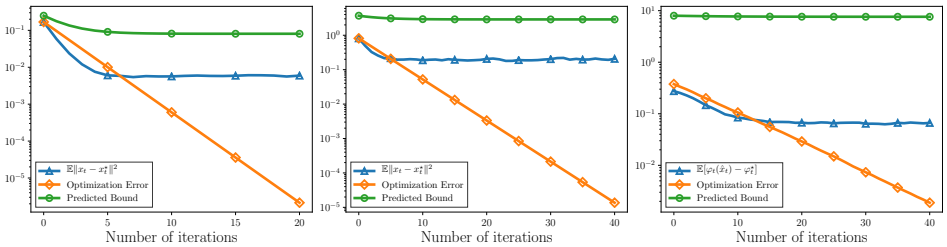

Figure 1: From left to right, *distance to minimum* for i) sparse vector least-squares estimation; ii) least-squares recovery; and *function gap* for iii) ii) least-squares recovery.

Moreover, the following bound on the asymptotic error holds:

$$\limsup_{t\to\infty} \mathbb{E}[\varphi_t(\hat{x}_t) - \varphi_t^\star] \lesssim \mu \left(\frac{\eta\sigma^2}{\mu} + \left(\frac{\Delta}{\mu\eta}\right)^2 \right).$$

A few comments about Theorem 3.5 are in order. First, each of the errors due to optimization, noise, and drift are known to be tight for the stochastic gradient method. The sum of the drift and noise terms coincides with μ times the analogous error term in Theorem 2.6, as expected. The optimization error in Theorem 3.5 is qualitatively different from the optimization error in Theorem 3.5, in that the term $\mathbb{E}\left[\varphi_t(x_0) - \varphi_t^\star + \frac{\mu}{4}\|x_0 - x_t^\star\|^2\right]$ can grow in time. Notice, however, that this term depends only on the evolution of the functions in time, and not on the iterates x_t nor the learning rate η. In particular, a byproduct of the theorem is that this quantity grows at most quadratically in time and therefore the entire optimization error tends to zero.

Our final result is an analogue of Theorem 3.5 that holds with high probability. Naturally, such a result should rely on light tail assumptions on the random variables $G_{i,t}$ and $\|z_t\|$ in Assumption 3.4. The result in full generality appears in the supplementary documents. Here, instead we state the guarantees under the assumption that $G_{i,t}$ and $\|z_t\|$ are uniformly bounded, which yields a simpler theorem statement and a better efficiency estimate. The theorem also for the first time requires z_i and $x_t^\star$ to be $\mathcal{F}_i$-independent for all $0 \le i < t$; this is a very reasonable assumption in light of Examples 2.1 and 2.2. The proof of Theorem 3.6 is based on combining the generalized Freedman inequality of [14] with a careful control of errors/bias due to the drift in one-step improvement estimates for the stochastic gradient method.

Theorem 3.6 (Function gap with high probability)**.** *Let $\{x_t\}$ be the iterates produced by Algorithm 1 with constant step-size $\eta \le 1/2L$. Fix $t \ge 1$, and suppose that there exist $\sigma, B \ge 0$ such that $\|z_i\| \le \sigma$ and $\sup_x \|\nabla f_t(x) - \nabla f_i(x)\| \le B(t-i)$ for all $0 \le i < t$. Suppose further that z_i and $x_t^\star$ are $\mathcal{F}_i$-independent for all $0 \le i < t$. Then for all $\delta \in (0, 1/e]$, the following bound holds with probability at least $1-\delta$:*

$$\begin{aligned}\varphi_t(\hat{x}_t) - \varphi_t^\star \le{}& (1 - \tfrac{\mu\eta}{2})^t \left(\varphi_t(x_0) - \varphi_t^\star + \tfrac{\mu}{4}\|x_0 - x_t^\star\|^2\right) \\ &+ c\left((1-\tfrac{\mu\eta}{2})^{t/2} \sqrt{(\|x_0 - x_0^\star\|^2 + t^2\Delta^2)\eta\mu\sigma^2} + \mu\left(\frac{\eta\sigma^2}{\mu} + \left(\frac{\Delta}{\mu\eta}\right)^2\right)\right) \log\left(\tfrac{1}{\delta}\right),\end{aligned}$$

where $c > 0$ is an absolute constant.

4 Numerical illustrations

We investigate the empirical behavior of our finite-time bounds on numerical examples with synthetic data. We consider examples of a) least-squares estimation of a sparse vector in Gaussian noise; b) least-squares recovery; and investigate the behavior of $\|x_t - x_t^\star\|^2$ and $\varphi_t(\hat{x}_t) - \varphi_t^\star$. The main findings are: 1) our finite-time bounds appear to have the correct scaling behavior with respect to η, σ, and Δ. 2) our high-probability bounds appear to have excellent coverage in Monte-Carlo simulations. We summarize our experimental settings and findings here. See also the Supplement.

Least-squares estimation of a sparse vector in Gaussian noise. We work here with a constrained formulation. Let B_1 denote the closed ℓ_1-ball in $\mathbf{R}^d$. Let $(x_t^\star)$ be a random walk in B_1 such that

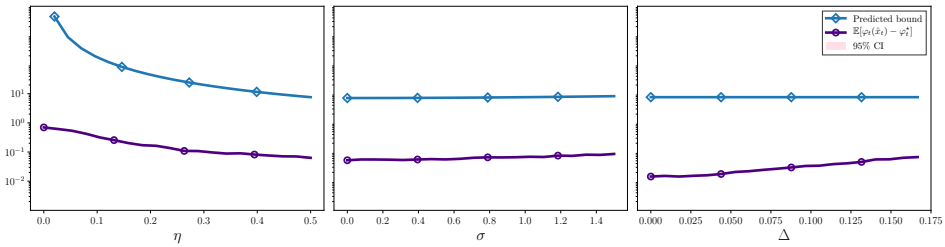

Figure 2: Theoretical vs. empirical scaling of function gap w.r.t. η, σ, and Δ for least-squares recovery.

$\|x_t^\star - x_{t+1}^\star\| \leq \Delta$ for all t. At each time t, sample $z_{t,1}, \ldots, z_{t,n}$ in B_1 such that $\frac{1}{n}\sum_{i=1}^n z_{t,i} = x_t^\star$ and $\|z_{t,i} - x_t^\star\| \leq \sigma$. The goal is to estimate using least-squares the sparse vectors $(x_t^\star)$.

$$\min_{x \in B_1} \frac{1}{2n} \sum_{i=1}^n \|x - z_{t,i}\|^2 = \min_{x \in \mathbf{R}^d} \varphi_t(x) := f_t(x) + r(x),$$

where $f_t(x) = \frac{1}{2n}\sum_{i=1}^n \|x - z_{t,i}\|^2$ and $r = \delta_{B_1}$; note that for each $\eta > 0$, $\mathrm{prox}_{\eta r}$ is the orthogonal projection onto B_1. We implement Algorithm 1 using gradient proxy $g_t = x_t - z_{t,i}$ at step t, where i is sampled uniformly from $\{1, \ldots, n\}$. We uniformly sample a vector in B_1 to fix as the initial x_0 in our simulations. We take $n = 15$, $d = 8$, $\sigma = 1/4$, and $\Delta = 1/20$; since f_t is 1-strongly convex and 1-smooth, this puts us in the low drift/noise setting $\frac{\Delta}{\sigma} < \sqrt{\frac{\mu}{16L^3}}$. To estimate $\mathbb{E}\|x_t - x_t^\star\|^2$ as well as confidence intervals for $\|x_t - x_t^\star\|^2$, we run 500 trials of Algorithm 1 with horizon $T = 25$, initiating the random walk $(x_t^\star)$ at the origin for each trial. The numerical results in Fig. 1-2 show that our bounds capture the correct scalings with respect to η, σ, and Δ both in distance to minimum and in function gap. See also Supplement.

Least-squares recovery. We work here with an un-regularized formulation. Let $(x_t^\star)$ be a random walk in $\mathbf{R}^d$, initialized at the origin, satisfying $\|x_t^\star - x_{t+1}^\star\| = \Delta$ for all t. Let $A \in \mathbf{R}^{n \times d}$, $C = \frac{\sigma^2}{n\|A\|_{\mathrm{op}}^2} I_n$, and $\mathcal{P}_t = \mathcal{N}(Ax_t^\star, C)$. The goal is to estimate using least-squares the vectors $(x_t^\star)$.

$$\min_{x \in \mathbf{R}^d} \mathop{\mathbb{E}}_{z \sim \mathcal{P}_t} \frac{1}{2}\|Ax - z\|^2 = \min_{x \in \mathbf{R}^d} f_t(x),$$

where $f_t(x) = \mathop{\mathbb{E}}_{z \sim \mathcal{P}_t} \frac{1}{2}\|Ax - z\|^2$. At step t we use gradient proxy $g_t = A^T(Ax_t - z)$, where $z \sim \mathcal{P}_t$. We uniformly sample a vector in the unit ℓ_2-ball to fix as the initial x_0 in our simulations, we randomly generate A via its singular value decomposition so that the minimal singular value of A is $\sqrt{\mu} = \sqrt{1/2}$ and the maximal singular value of A is $\sqrt{L} = 1$, and we take $n = 10$, $d = 5$, $\sigma = 1$, and $\Delta = 1/6$; since f_t is μ-strongly convex and L-smooth, this puts us in the low drift/noise setting $\frac{\Delta}{\sigma} < \sqrt{\frac{\mu}{16L^3}}$. To estimate $\mathbb{E}\|x_t - x_t^\star\|^2$ and $\mathbb{E}[\varphi_t(\hat{x}_t) - \varphi_t^\star]$ as well as confidence intervals for $\|x_t - x_t^\star\|^2$ and $\varphi_t(\hat{x}_t) - \varphi_t^\star$, we run 250 trials of Algorithm 1 with horizon $T = 40$, initiating the random walk $(x_t^\star)$ at the origin for each trial. Here again the numerical results in Fig. 1-2 show that our bounds capture the correct scalings with respect to η, σ, and Δ in function gap.

Conclusion We presented finite-time efficiency estimates in expectation and in high-probability for the tracking error of the proximal stochastic gradient method under time drift. Through our investigation of the interplay between the learning rate, the noise variance in the gradient oracle, and the strength of the time drift, we uncovered two regimes of interest, that suggest different choices of learning rates. The high probability result extend recent results to stochastic optimization under time drift. The numerical illustrations on synthetic data show the excellent coverage of the high probability guarantees.

Broader Impacts Discussion in this section is mostly speculative, as the primary goal of this work is a theoretical study. The results presented here could enable a deeper understanding of stochastic learning under time drift. We acknowledge the major issues of deploying machine learning "in the wild" without very careful and prepared anticipation of the risks. We also acknowledge the limitations of our theory, including changing environments that evolve in a dependent manner to the learning algorithm, which would be an interesting venue for future work.

References

[1] A. Agarwal, O. Chapelle, M. Dudík, and J. Langford. A reliable effective terascale linear learning system. *J. Mach. Learn. Res.*, 15(1):1111–1133, 2014.

[2] F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[3] P. Bartlett, V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory-COLT 2008*, pages 335–342. Omnipress, 2008.

[4] P. L. Bartlett, S. Ben-David, and S. R. Kulkarni. Learning changing concepts by exploiting the structure of change. *Machine Learning*, 41(2):153–174, 2000.

[5] O. Besbes, Y. Gur, and A. Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[6] L. Bottou. Stochastic learning. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176 of *Lecture Notes in Computer Science*, pages 146–168. Springer, 2003.

[7] L. Bottou. Stochastic gradient descent tricks. In G. Montavon, G. B. Orr, and K. Müller, editors, *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer, 2012.

[8] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

[9] C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1. JMLR Workshop and Conference Proceedings, 2012.

[10] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020.

[11] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[12] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[13] L. Guo and L. Ljung. Exponential stability of general tracking algorithms. *IEEE Transactions on Automatic Control*, 40(8):1376–1387, 1995.

[14] N. J. Harvey, C. Liaw, Y. Plan, and S. Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *Conference on Learning Theory*, pages 1579–1613. PMLR, 2019.

[15] N. J. A. Harvey, C. Liaw, Y. Plan, and S. Randhawa. Tight analyses for non-smooth stochastic gradient descent. In A. Beygelzimer and D. Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1579–1613, Phoenix, USA, 25–28 Jun 2019. PMLR.

[16] E. Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019.

[17] E. Hazan and S. Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

[18] E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, pages 393–400, 2009.

[19] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

[20] D. Hsu, S. Kakade, T. Zhang, et al. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012.

[21] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

[22] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*, volume 35 of *Applications of Mathematics*. Springer, 1997.

[23] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

[24] H. Lang, L. Xiao, and P. Zhang. Using statistics to automate stochastic optimization. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9536–9546, 2019.

[25] J. Lin and L. Rosasco. Optimal learning for multi-pass stochastic gradient methods. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[26] P. M. Long. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354, 1999.

[27] L. Madden, S. Becker, and E. Dall'Anese. Bounds for the tracking error of first-order online optimization methods. *Journal of Optimization Theory and Applications*, 189(2):437–457, 2021.

[28] A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

[29] A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019. PMLR, 2013.

[30] A. H. Sayed. *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[31] R. E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 2012.

[32] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for machine learning*. MIT Press, 2012.

[33] N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[34] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[35] C. Wilson, V. V. Veeravalli, and A. Nedić. Adaptive sequential stochastic optimization. *IEEE Transactions on Automatic Control*, 64(2):496–509, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [N/A]
- Did you include the license to the code and datasets? [N/A]
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See "Broader Impacts" on page 9.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? **[TODO]**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[TODO]**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[TODO]**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? **[TODO]**
   (b) Did you mention the license of the assets? **[TODO]**
   (c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? **[TODO]**
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? **[TODO]**

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **[TODO]**