2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Stochastic Fast Gradient for Tracking

Dmitry Kosaty[1], Alexander Vakhitov[2], Oleg Granichin[3], and Ming Yuchi[4]

Abstract— **In recent applications, first-order optimization methods are often applied in the non-stationary setting when the minimum point is drifting in time, addressing a so-called parameter tracking, or non-stationary optimization (NSO) problem. In this paper, we propose a new method for NSO derived from Nesterov's Fast Gradient. We derive theoretical bounds on the expected estimation error. We illustrate our results with simulation showing that the proposed method gives more accurate estimates of the minimum points than the unmodified Fast Gradient or Stochastic Gradient in case of deterministic drift while in purely random walk all methods behave similarly. The proposed method can be used to train convolutional neural networks to obtain super-resolution of digital surface models.**

I. INTRODUCTION

The Fast Gradient (Accelerated Gradient) algorithm is optimal among the gradient-only methods in optimization of strongly convex functions [1]. B.T. Polyak analyzed a method of a similar type in a seminal 1964 publication [2], and called it the Heavy Ball method. When gradient measurements are not exact but corrupted with additive noise, in the so-called stochastic optimization (SO) problem, in [3] it is argued that the Heavy Ball method has the same convergence rate as Stochastic Gradient Descent (SGD). The SGD is asymptotically optimal for the SO. However, after a finite number of iterations these methods may have different accuracy of estimates. Gradient-only methods receive much attention of the engineering and computer science researchers willing to solve large-dimensional problems, where second-order methods are too expensive, e.g. for deep learning. To make one step of an iterative learning method faster, in deep learning one commonly chooses only a small random sample of training examples ('batch') to compute the gradient, making a problem similar to stochastic optimization. The Fast Gradient is successfully applied for the optimization of deep learning architectures [4–6].

With the increase of data uploaded to the web every day, attention in the artificial intelligence community is attracted to online learning problems. In this case, the data distribution becomes non-stationary [7–9]. Online learning of the deep neural network is a good motivational example of NSO for a highly nonlinear function.

Classical stochastic approximation algorithms such as Robbins-Monro procedure use diminishing gain sequence to estimate the unknown parameters, meaning that with every new step less weight is given to recent gradient measurements. To apply them in the NSO setting, we need to change the diminishing gain sequence to a fixed constant gain emphasizing recent observations. There are different parameter drift models for the nonstationary optimization, starting with a random walk and Kalman-type state evolution, as well as other models [10–21]. In this paper, we choose a general model of unknown but bounded parameter drift similar to [16], [17], [21], which includes random walk-type drift [10], [13], deterministic evolution and many intermediate formulations.

Our results show the existence of a finite bound on average mean squared estimation error. The simulation illustrates that the proposed method with an appropriate choice of parameters in all the considered drift scenarios has same (random drift) or significantly higher (deterministic linear and nonlinear drift) accuracy compared to the SGD. Several approaches for proving the convergence are based on the ordinary differential equations (ODE) analysis, emphasizing asymptotic rates and not giving any prediction of the estimation error bounds after a finite number of steps [19], [20], [22]. In this paper, we adopt a different an approach inspired by [23]. The next section of the paper describes the problem statement and presents the new SFGT algorithm. The third section contains conditions under which the error of the minimum estimates provided by the SFGT method stays within certain bounds. In the fourth section we report simulation results, and then formulate the conclusions.

II. PROBLEM AND ALGORITHM FORMULATION

We need to find a sequence of parameters $\theta_n \in \mathbb{R}^q$ which minimize corresponding differentiable loss functions:

$$\text{Find} \quad \theta_n = \text{Argmin}_{\theta\in\Theta} f_n(\theta), \quad \forall n \in \mathbb{N} \tag{1}$$

Here and further n is a time instance.

Sometimes (1) is called parameter tracking problem. We denote the conditional expectation w.r.t. σ-algebra defined by $\hat{\theta}_0, \ldots, \hat{\theta}_{n-1}$ as $\mathbb{E}_n$. We consider gradient measurements $Y_n(\theta)$ corrupted by additive noise $\xi_n \in \mathbb{R}^q$ as the only available information:

$$Y_n(\theta) = \nabla f_n(\theta) + \xi_n, \quad n \in \mathbb{N} \tag{2}$$

This work was supported by the National Key Research and Development Program of China (2016YFE0203900) with cooperation by the Russian Foundation for Basic Research (project 17-51-53053).

[1]Dmitry Kosaty is with Faculty of Mathematics and Mechanics, Saint Petersburg State University, 7-9, Universitetskaya Nab., Saint Petersburg, 199034, Russia dkosaty@gmail.com

[2]Alexander Vakhitov is with Samsung AI Center, 10, Butirskiy val, Moscow, 125047, Russia a.vakhitov@samsung.com

[3]Oleg Granichin is with the Saint Petersburg State University (Faculty of Mathematics and Mechanics, and Research Laboratory for Analysis and Modeling of Social Processes), 7-9, Universitetskaya Nab., Saint Petersburg, 199034, Russia o.granichin@spbu.ru

[4]Ming Yuchi is with the School of Life Science and Technology Huazhong university of Science and Technology, 1037 Luoyu Road, Wuhan, Hubei, 430074, China m.yuchi@hust.edu.cn

The algorithm proposed in this paper provides a sequence of estimates $\{\hat{\theta}_n\}_{n=0}^{\infty}$ solving the following problem:

$$\begin{aligned} \text{Find} \quad \hat{\theta}_n \quad \text{s.t.} \quad \exists N, C<\infty: \quad \forall n>N \\ \mathbb{E}\|\hat{\theta}_n-\theta_n\|^2 \le C \end{aligned} \tag{3}$$

We denote $f_n^* = f_n(\theta_n)$. We assume the following properties of the functions f_n.

Assumption 1. Functions f_n have a common Lipshitz constant $L>0$ and strong convexity constant $\mu>0$:

$$\begin{aligned} \forall x \in \mathbb{R}^q \quad \|\nabla f_n(x)\| \le L\|x-\theta_n\|, \\ \langle \nabla f_n(x), x-\theta_n\rangle \ge \mu\|x-\theta_n\|^2 \end{aligned} \tag{4}$$

Assumption 2. For every $n \ge 0$, drift is bounded as

$$\|f_n(x)-f_{n+1}(x)\| \le a\|\nabla f_n(x)\|+b, \quad f_n^*=f^* \tag{5}$$

$$\|\nabla f_{n+1}(x)-\nabla f_n(x)\| \le c \tag{6}$$

Note 1. For a quadratic function $f_n(x)=(x-\theta_n)^T Q(x-\theta_n)$ with positive definite Q and minimum point drift satisfying $\|\theta_n-\theta_{n-1}\| \le d$, $\nabla f_n(x)=2Q(x-\theta_n)$, Assumption 1 holds with $L=2\|Q\|$, $\mu=2\lambda_{min}(Q)$, Assumption 2 holds with $a=2d$, $b=\|Q\|d^2$ and $c=2\|Q\|d$.

Note 2. By virtue of Assumption 2, the minimum point drift can be bounded as follows:

$$\begin{aligned} \|\theta_{n+1}-\theta_n\| \le \mu^{-1}\|\nabla f_n(\theta_{n+1})\| = \\ = \mu^{-1}\|\nabla f_{n+1}(\theta_{n+1})\| - \|\nabla f_n(\theta_{n+1})\| \le \mu^{-1}c \end{aligned}$$

Assumption 3. The noise ξ_n is zero mean and has a bounded variance σ^2.

Note 3. It is possible to use a weaker assumption $\mathbb{E}\|\xi_n\|^r \le \sigma^r$, $r \in (1,2)$ analogously to [24].

To solve the problem defined by (3) with observation model (2) under Assumptions 1–3 we propose the Stochastic Fast Gradient for Tracking (SFGT) algorithm, which has the following form:

1) Choose $\hat{\theta}_0 \in \mathbb{R}^q$, $\gamma_0>0$. Set $v_0=\hat{\theta}_0$. Choose $h>0$, $\eta \in (0,\mu)$, $\alpha_x \in (0,1)$ so that α_n satisfying the inequality (7) can always be found. Define $H_1 = h - \frac{h^2 L}{2}$.
2) n-th iteration ($n \ge 0$):
   a) Find $\alpha_n \in [\alpha_x, 1)$ so that
   $$H_1 - \frac{\alpha_n^2}{2\gamma_{n+1}} > 0. \tag{7}$$
   b) Let $\gamma_{n+1}=(1-\alpha_n)\gamma_n+\alpha_n(\mu-\eta)$.
   c) Choose $x_n = \frac{1}{\gamma_n+\alpha_n(\mu-\eta)}\Big(\alpha_n\gamma_n v_n+\gamma_{n+1}\hat{\theta}_n\Big)$ and compute $Y_n(x_n)$.
   d) Find a new estimate $\hat{\theta}_{n+1}$: $\hat{\theta}_n = x_n - hY_n(x_n)$.
   e) Set $v_{n+1}=\frac{1}{\gamma_n}\Big[(1-\alpha_n)\gamma_n v_n+\alpha_n(\mu-\eta)x_n - \alpha_n Y_n(x_n)\Big]$.

We show here that the inequality (7) can always be fulfilled by some choice of parameters, further we identify the best strategies to set the parameters.

Note 4. The combination of parameters satisfying all conditions does exist. One can choose $0 \le \eta \le \mu$, then $\alpha_x < \sqrt{\frac{\mu-\eta}{L}}$ and $\gamma_0 > \mu-\eta$, $h=L^{-1}$. Then $\gamma_n > \mu-\eta$, and the choice of α_x assures that the inequality (7) holds for α_x.

III. MAIN RESULT

Theorem 1: The problem (3) is solved by the SFGT algorithm with

$$C = \frac{2}{\mu}D_\infty \tag{8}$$

where

$$\begin{aligned} D_\infty = \alpha_x^{-1}\Big[\frac{2a+hc}{4\epsilon}+2b+ \\ +(1-\alpha_x)(b+A_\infty c)+ \\ +h^2\frac{L}{2}\sigma^2+\frac{c^2}{2\eta}\Big] \end{aligned} \tag{9}$$

for $\Gamma = max_{n\ge 0}\gamma_n$, $\epsilon \in \Big(0, \frac{1}{a(1+\alpha_x)+hc}\Big(H_1-\frac{\alpha_x^2}{2\Gamma}\Big)\Big]$ and α_x, η, h chosen in the algorithm.

The estimation error after a finite number of iterations is bounded as:

$$\mathbb{E}_n f_n(\hat{\theta}_n) - f_n^* \le \prod_{i=1}^{n}(1-\alpha_n)(\phi_0(\theta_0)-f^*+\Phi)+D_n$$

where $\phi_0(x)=f_0(\hat{\theta}_0)+\frac{\gamma_0}{2}\|x-v_0\|^2$, $\Phi=\frac{\gamma_0 c^2}{2\mu^2}$, $\{\alpha_n\}_{n=0}^{\infty}$, $\{\lambda_n\}_{n=0}^{\infty}$, $\{A_n\}_{n=0}^{\infty}$ and $\{Z_n\}_{n=0}^{\infty}$ are sequences defined as

$$\alpha_n \in [\alpha_x, 1), \quad \lambda_0=1, \quad \lambda_{n+1}=(1-\alpha_n)\lambda_n \tag{10}$$

$$\begin{aligned} A_0=0, \quad A_{n+1}=(1-\alpha_n)((1-\lambda_n)a+A_n), \\ Z_n=(1-\lambda)(b+ac)+A_n c, \\ D_0=0, \quad D_{n+1}=(1-\alpha_n)D_n+\frac{a(1+\alpha_n)+hc}{4\epsilon}+ \\ +(1+\alpha_n)b+(1-\alpha_n)Z_n+ \\ +h^2\frac{L}{2}\sigma^2+\frac{\alpha_n c^2}{2\eta} \end{aligned}$$

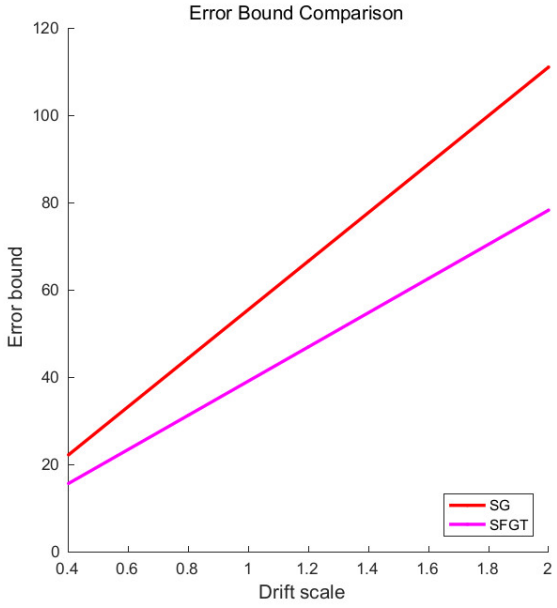
Note 5. The bound (8), (9) can be optimized by the algorithm parameters, focusing on the optimality of the bound for large n. We see that the optimal value for ϵ is the maximal one. As we see from the recursion for γ_n defined in the algorithm, $\gamma_n \to \mu-\eta$, so in (9) we can set $\Gamma = \mu-\eta$ and choose optimal $\epsilon = \frac{1}{a(1+\alpha_x)+hc}\Big(H_1-\frac{\alpha_x^2}{2(\mu-\eta)}\Big)$, getting (11), which we need to minimize by α_x, h, η.

$$\begin{aligned} D_\infty = \alpha_x^{-1}\Big[\frac{a^2(1+\alpha_x)}{2H_1-(\mu-\eta)^{-1}\alpha_x^2}+ \\ +(3-\alpha_x)b+ac+h^2\frac{L}{2}\sigma^2+\frac{c^2}{2\eta}\Big] \end{aligned} \tag{11}$$

The proof is inspired by [23]. In the proof we use several lemmas proved in the Appendix.

Fig. 1 shows the comparison of theoretically predicted bounds for the SG and SFGT algorithms. The bound for the gradient descent with constant step size for tracking is taken from the paper [21], and because it is devoted to parameter trackng in deterministic setting, we set the noise standard deviation $\sigma_v = 0$. We evaluate the bounds on a

Fig. 1: Comparison of the theoretical bound for the parameter tracking for gradient descent [21] and for SFGT derived in this paper



particular problem, same as deterministic linear drift case in the following section, where the function and the drift law are described. We vary the drift scale and show the predicted asymptotic bound on the estimation error norm, $\|\hat{\theta}_n - \theta_n\|$. We choose the parameters of the methods (step size in case of gradiend descent, α_x, η, h in SFGT) optimizing the predicted error bound. As we see, theoretically SFGT method outperforms the gradient descent.

IV. SIMULATION

We compare the proposed SFGT method with earlier published method aiming at the same problem, Stochastic Gradient Descent (SG) analyzed for example in [19], [21], Accelerated Gradient (AG) [23], and Kalman filter as a baseline. Kalman filter benefits from additional information about the Hessian matrix compared to the other methods.

We choose $q = 10$, i.e. 10-dimensional space, the function to minimize is $f(x, \theta_n)$:

$$f(x, \theta_n) = \frac{1}{2}(x - \theta_n)^T Q (x - \theta_n) \tag{12}$$

where $Q = diag\{1, 2, \ldots, 10\}$ is diagonal matrix and the drift of minimum point is defined as

$$\begin{aligned} \theta_0 &= 0 \\ \theta_n &= \theta_{n-1} + t_n \end{aligned} \tag{13}$$

We denote the unit matrix as I. We consider three drift scenarios, all parameterized by drift length d: random (defined as (14) by choosing drift direction from random normal distribution), linear (defined as (15)) or nonlinear defined as (16).

$$\bar{t}_n \in N(0, I), \quad t_n = d\frac{\bar{t}_n}{\|\bar{t}_n\|} \tag{14}$$

$$\bar{t}_n = d\frac{\bar{t}}{\|\bar{t}\|} \tag{15}$$

$$\begin{aligned} m(n) &= \mod(n, 100), \\ \zeta_n &= 0.01 m(n) \zeta_1 + (1 - 0.01 m(n)) \zeta_2 \\ \bar{t}_n &= d\frac{\zeta_n}{\|\zeta_n\|} \end{aligned} \tag{16}$$

The gradient measurement is defined as

$$g(x, \theta_n) = \nabla f(x, \theta_n) + \sigma \xi_n \tag{17}$$

and ξ_n is a mean-zero Gaussian vector with i.i.d. components with unit variance.

We chose to use constant $\alpha_n = \alpha_x$, $\gamma_0 = \mu - \eta$ in the SFGT method. The step-size of the Stochastic Gradient is denoted as h. The Fast (Accelerated) Gradient method is implemented as described in [23], section 2.2.1, with step size h. The Kalman filter is used with unit state evolution matrix, initial error covariance matrix $\sigma_0 I$, state noise covariance $\sigma_s I$, measurement covariance $\sigma_m I$ and measurement matrix Q. The parameters of the methods unless otherwise stated are chosen by exhaustive search and set to the values delivering lowest root mean squared error (RMSE) at the 500th iteration, the values are given in the Table I. In all cases $\theta_0 = 0$, $\hat{\theta}_0 = (35, 15, 35, \ldots, 15)^T$.

Fig. 2 shows parts of the trajectories of the methods and the plots of the RMSE for the minimum estimation for each scenario, for the SFGT_Opt we show also the error bars equal to error variances. We show two plots for the SFGT method, one with theoretically optimal parameters chosen by minimizing the bound (9) labelled SFGT_T, and another one with parameters chosen by exhaustive search SFGT_Opt as it was done with other methods. In all cases we averaged over 100 runs of each method.

Fig. 3 shows the dependency of RMSE on 1000-th iteration on drift and measurement noise scale with the other parameter fixed at $d = 0.1$, $\sigma = 0.1$. We see that while for the random drift all methods behave similarly, for the linear and non-linear deterministic drift SFGT_Opt outperforms other methods except the Kalman filter, and the method with theoretically optimal parameters SFGT_T gives estimation errors slightly worse than SG, but it is worth noting that for the latter simulation based exhaustive search for the best parameter was used while SFGT_T parameters were chosen solely using the analytical formula derived in this paper. The unmodified FG method has lower accuracy than SG and SFGT in case of deterministic drift.

V. CONCLUSIONS

In this paper, we have given bounds on the estimation errors of the Stochastic Gradient with Momentum algorithm in the Nonstationary Optimization Problem. From the theoretical results we see that a new Fast Gradient-type method can be applied in this problem. This is illustrated with numerical simulation. The proposed method can be used to train convolutional neural networks to obtain super-resolution of digital surface models.

TABLE I: The parameters of the methods used in comparison, L for linear drift, R for random, N for nonlinear

Scenario (σ, d)	SG (h)	FG (h) (σ_0, σ_m, σ_s)	Kalman (h, α_x, η)	SFGT_Opt (h, α_x, η)	SFGT_T (h, α_x, η)
Figure 2.					
L, (0.10, 1.0)	0.19	0.10	2.0, 0.20, 0.6	0.1, 0.09, 0.9	0.08, 0.21, 0.3
N, (0.10, 1.0)	0.19	0.10	2.0, 0.02, 0.2	0.1, 0.09, 0.9	0.08, 0.21, 0.3
R, (0.10, 1.0)	0.15	0.09	0.6, 0.20, 0.6	0.1, 0.20, 0.5	0.08, 0.21, 0.3
Figure 3.					
L, (0.10, 0.0)	0.02	0.00	0.00, 0.02, 0.90	0.03, 0.01, 0.80	0.01, 0.13, 0.05
L, (0.10, 0.5)	0.19	0.10	0.90, 0.12, 0.80	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (0.10, 1.0)	0.19	0.10	2.00, 0.20, 0.60	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (0.10, 1.5)	0.19	0.10	1.50, 0.20, 0.10	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (0.10, 2.0)	0.19	0.10	4.00, 0.14, 0.70	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (0.14, 1.0)	0.19	0.10	0.80, 0.24, 0.90	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (0.37, 1.0)	0.19	0.10	0.40, 0.37, 1.00	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (1.00, 1.0)	0.19	0.10	0.40, 0.80, 0.10	0.10, 0.09, 0.90	0.08, 0.21, 0.30
L, (2.72, 1.0)	0.16	0.10	0.60, 1.63, 1.00	0.07, 0.06, 0.90	0.08, 0.21, 0.30
L, (7.39, 1.0)	0.07	0.03	0.40, 4.43, 0.40	0.03, 0.03, 0.80	0.08, 0.21, 0.30
R, (0.10, 0.0)	0.02	0.00	0.00, 0.06, 0.50	0.03, 0.04, 0.20	0.01, 0.13, 0.05
R, (0.10, 0.5)	0.12	0.10	0.50, 0.20, 0.40	0.10, 0.16, 0.70	0.08, 0.21, 0.30
R, (0.10, 1.0)	0.15	0.09	0.60, 0.20, 0.60	0.10, 0.20, 0.50	0.08, 0.21, 0.30
R, (0.10, 1.5)	0.15	0.09	1.80, 0.10, 0.30	0.10, 0.23, 0.20	0.08, 0.21, 0.30
R, (0.10, 2.0)	0.12	0.10	3.20, 0.04, 0.90	0.10, 0.16, 0.70	0.08, 0.21, 0.30
R, (0.14, 1.0)	0.16	0.10	0.60, 0.22, 0.10	0.10, 0.22, 0.40	0.08, 0.21, 0.30
R, (0.37, 1.0)	0.15	0.08	0.20, 0.29, 0.80	0.10, 0.09, 0.90	0.08, 0.21, 0.30
R, (1.00, 1.0)	0.13	0.09	0.20, 0.80, 0.70	0.10, 0.14, 0.70	0.08, 0.21, 0.30
R, (2.72, 1.0)	0.12	0.08	0.60, 4.35, 0.30	0.07, 0.02, 0.30	0.08, 0.21, 0.30
R, (7.39, 1.0)	0.04	0.05	0.40, 8.87, 0.80	0.07, 0.02, 0.90	0.08, 0.21, 0.30

Fig. 2: Trajectory part and RMSE (with variance bar for SFGT_Opt) for directional, random and nonlinear drift scenarios (see text)

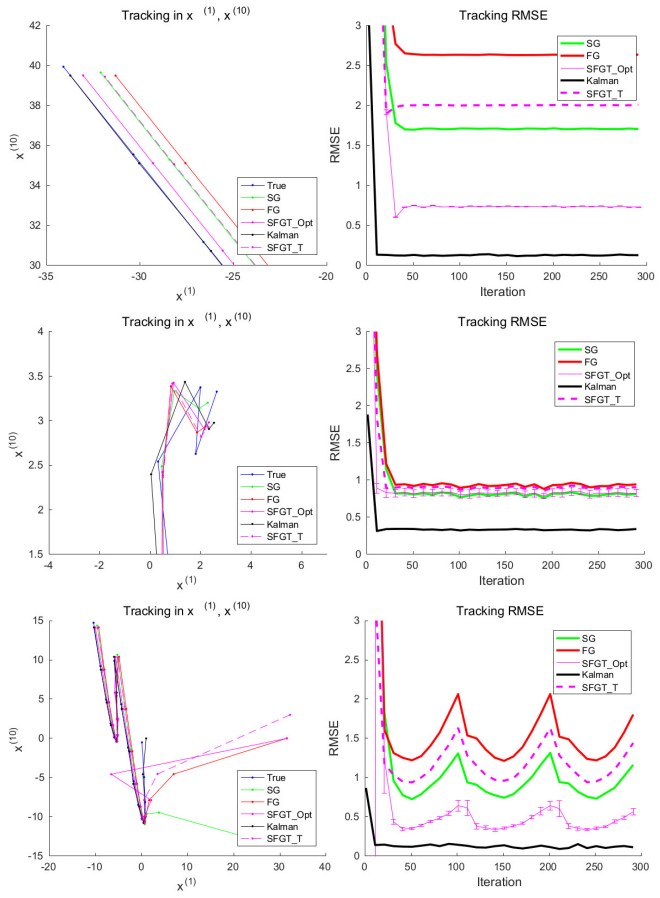
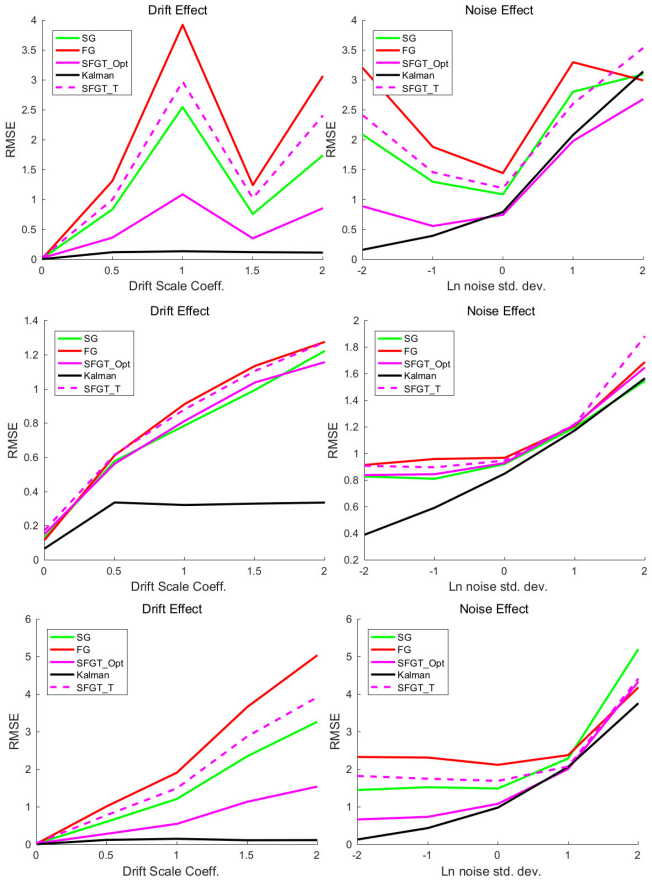


Fig. 3: Dependency on drift and noise scale for directional, random and nonlinear drift scenarios (see text)



VI. Appendix

Definition. A pair of a sequence $\{\lambda_n\}_{n=0}^{\infty}, \lambda_n \geq 0$, and a sequence of functions $\{\phi_n(x)\}_{n=0}^{\infty}$ are called an A_n, Φ-bounded estimate sequence for functions $\{f_n(x)\}$ if

$$\lambda_n \to 0, \tag{18}$$

and there exist a sequence $\{A_n\}_{n=0}^{\infty}$, $A_n \in \mathbb{R}$ and a constant $\Phi < \infty$ such that, denoting $\tilde{\phi}_{0,n}(x) = \phi_0(x) - \phi_0(\theta_n) + \phi_0(\theta_0)$,

$$\begin{gathered}\mathbb{E}\phi_n(x) \leq (1-\lambda_n)f_n(x)+ \\ +A_n\|\nabla f_n(x)\| + \lambda_n(\tilde{\phi}_{0,n}(x) + \Phi)\end{gathered} \tag{19}$$

Lemma 1. *Let $\{x_n\}_{n=0}^{\infty}$ be an arbitrary sequence in $\mathbb{R}^q$, $\{\alpha_n\}_{n=0}^{\infty}, \{\lambda_n\}_{n=0}^{\infty}$ be arbitrary sequences in $\mathbb{R}$ such that (10) holds, $\{A_n\}_{n=0}^{\infty}$, $\{Z_n\}_{n=0}^{\infty}$ be sequences in $\mathbb{R}$ defined as $A_0 = 0$, $A_{n+1} = (1-\alpha_n)[(1-\lambda_n)a + A_n]$, $\Phi = \frac{\gamma_0 c^2}{2\mu^2}$, $Z_0 = 0$, $Z_{n+1} = (1-\lambda_{n+1})(b+ac) + A_{n+1}c$, $\{\phi_n(x)\}_{n=0}^{\infty}$ be a sequence of functions defined using $\eta \in (0, \mu]$, $\gamma_0 > 0$ and $r(x_n) = f_n(x_n) - \frac{c^2}{2\eta} - a\|\nabla f_n(x_n)\| - b$ as follows*

$$\begin{gathered}\phi_0(x) = \phi_0^* + \frac{\gamma_0}{2}\|x - x_0\|^2 \\ \phi_{n+1}(x) = (1-\alpha_n)(\phi_n(x) - Z_n) + \alpha_n[r(x_n) \\ +\langle Y_n(x_n), x - x_n\rangle + (\frac{\mu}{2} - \frac{\eta}{2})\|x - x_n\|^2]\end{gathered} \tag{20}$$

where $\mathbb{E}Y_n(x_n) = \nabla f_n(x_n)$.

Then $\{\phi_n\}_{n=0}^{\infty}$ forms an A_n, Φ-bounded estimate sequence for $\{f_n\}_{n=0}^{\infty}$.

Note. Because $\alpha_j \geq \alpha_x > 0$, $\{A_n\}, \{Z_n\}$ are bounded uniformly in n: $A_n < A_\infty < \infty$, $Z_n < Z_\infty < \infty$, and $A_\infty = \frac{a}{\alpha_x}$, $Z_\infty = b + \frac{ac(1+\alpha_x)}{\alpha_x}$.

Proof. We use induction by $n \geq 0$. Using the fact that $A_0 = 0$, $\lambda_0 = 1$ we see that the base holds:

$$\phi_0(x) = \tilde{\phi}_{0,0}(x) \leq \tilde{\phi}_{0,0}(x) + \Phi$$

Now we assume that the statement is valid for n and prove it for $n+1$. We have

$$\begin{gathered}\mathbb{E}\phi_{n+1}(x) \leq \alpha_n f_{n+1} + (1-\alpha_n)(\mathbb{E}\phi_n(x) - Z_n) \leq \\ \alpha_n f_{n+1} + (1-\alpha_n)((1-\lambda_n)f_n(x) + \lambda_n\tilde{\phi}_{0,n}(x)+ \\ +A_n\|\nabla f_n(x)\| - Z_n)\end{gathered}$$

As long as $1-\lambda_{n+1} = 1-(1-\alpha_n)\lambda_n = (1-\alpha_n)(1-\lambda_n)+\alpha_n$

$$\mathbb{E}\phi_{n+1}(x) \leq (1-\lambda_{n+1})f_{n+1}(x)+\lambda_{n+1}\tilde{\phi}_{0,n}(x)+(1-\alpha_n)\rho(x)$$

$$\rho(x) = (1-\lambda_n)(f_n(x) - f_{n+1}(x)) + A_n\|\nabla f_n(x)\| - Z_n$$

Using Assumption 2, we derive

$$\begin{gathered}\tilde{\phi}_{0,n}(x) - \tilde{\phi}_{0,n+1}(x) \leq \\ \leq |\phi_0(\theta_n) - \phi_0(\theta_{n+1})| \leq \\ \leq \frac{\gamma_0}{2}\|\theta_n - \theta_{n+1}\|^2 \leq \frac{\gamma_0 c^2}{2\mu^2}\end{gathered}$$

$$\begin{gathered}\rho(x) \leq (1-\lambda_n)(a\|\nabla f_{n+1}(x)\| + b)+ \\ +A_n\|\nabla f_{n+1}(x)\| + A_n c + (1-\lambda_n)ac - Z_n\end{gathered}$$

Now we get $(1-\alpha_n)[(1-\lambda_n)a + A_n]\|\nabla f_{n+1}(x)\| + (1-\lambda_n)(b+ac) + A_n c - Z_n = A_{n+1}\|\nabla f_{n+1}(x)\|$, by definition of Z_n and A_n. We have

$$A_{n+1} = (1-\alpha_n)[(1-\lambda_n)a+A_n] \leq (1-\alpha_x)(a+A_n) \leq \frac{a}{\alpha_x}$$

$$Z_{n+1} = (1-\lambda_{n+1})(b+ac) + A_{n+1}c \leq b + ac\frac{1+\alpha_x}{\alpha_x}$$

Lemma 2. *The functions ϕ_n defined by (20) can be expressed in form $\phi_n(x) = \phi_n^* + \frac{\gamma_n}{2}\|x - v_n\|^2$, with*

$$\begin{gathered}v_{n+1} = \gamma_{n+1}^{-1}((1-\alpha_n)\gamma_n v_n + \alpha_n(\mu-\eta)x_n - \alpha_n Y_n(x_n)) \\ \gamma_{n+1} = (1-\alpha_n)\gamma_n + \alpha_n(\mu-\eta) \\ \phi_{n+1}^* = (1-\alpha_n)(\phi_n^* - Z_n) - \frac{\alpha_n^2}{2\gamma_{n+1}}\|Y_n(x_n)\|^2+ \\ +\alpha_n\frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}}(\frac{(\mu-\eta)}{2}\|x_n-v_n\|^2-\langle x_n-v_n, Y_n(x_n)\rangle)+ \\ +\alpha_n r(x_n)\end{gathered}$$

Proof. Expression for the γ_n follows from taking the second derivative of the equation (20). If we take a gradient of ϕ_{n+1} and equate it to 0, we get:

$$(1-\alpha_n)\gamma_n(v_{n+1}-v_n)+\alpha_n Y_n(x_n)+\alpha_n(\mu-\eta)(v_{n+1}-x_n) = 0$$

which leads to an expression for v_{n+1}. Substituting x_n into (20), we get

$$\phi_{n+1}(x_n) = (1-\alpha_n)(\phi_n(x_n) - Z) + \alpha_n r(x_n) \tag{21}$$

In the same time, we have

$$\phi_{n+1}(x_n) = \phi_{n+1}^* + \frac{\gamma_{n+1}}{2}\|x_n - v_{n+1}\|^2 \tag{22}$$

Using the obtained expression for v_{n+1}, we get

$$x_n - v_{n+1} = (\frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}})(x_n - v_n) + \frac{\alpha_n}{\gamma_{n+1}}\nabla f_n(x_n)$$

so that

$$\begin{gathered}\frac{\gamma_{n+1}}{2}\|x_n - v_{n+1}\|^2 = \frac{\gamma_{n+1}}{2}(\frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}})^2\|x_n - v_n\|^2+ \\ +2\frac{\gamma_{n+1}}{2}(\frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}})\frac{\alpha_n}{\gamma_{n+1}}\langle x_n - v_n, \nabla f(x_n)\rangle+ \\ +\frac{\gamma_{n+1}}{2}(\frac{\alpha_n}{\gamma_{n+1}})^2\|\nabla f_n(x_n)\|^2\end{gathered}$$

Subtracting (22) from (21), we get a following coefficient of $\|x_n - v_n\|^2$:

$$\begin{gathered}(1-\alpha_n)\frac{\gamma_n}{2} - \frac{\gamma_{n+1}}{2}(\frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}})^2 = \\ = (1-\alpha_n)\frac{\gamma_n}{2} - \frac{1}{2}\frac{(1-\alpha_n)^2\gamma_n^2}{\gamma_{n+1}} = \\ = (1-\alpha_n)\frac{\gamma_n}{2}(1 - \frac{(1-\alpha_n)\gamma_n}{\gamma_{n+1}}) = \\ = \frac{1}{2}\frac{(1-\alpha_n)\alpha_n(\mu-\eta)\gamma_n}{\gamma_{n+1}}\end{gathered}$$

and we get an expression for ϕ_{n+1}^*.

Lemma 3. *(See [23], 2.2.1). If $\{\lambda_n\}, \{\phi_n(x)\}$ form a A_n, Φ-bounded estimate sequence for functions $\{f_n(x)\}$ and for some sequence $\{theta_n\}_{n=0}^{\infty}$ in $\mathbb{R}^q$, $\{D_n\}_{n=0}^{\infty}$ in $\mathbb{R}$, $D_n \geq 0$, $D_n < D_\infty < \infty$ the following inequalities hold for all $n \geq 0$:*

$$\mathbb{E}_n f_n(\theta_n) \leq \phi_n^* + D_n = \min_{x \in \mathbb{R}^q} \phi_n(x) + D_n \tag{23}$$

then

$$\begin{gathered} \mathbb{E}_n f_n(\theta_n) - f_n^* \leq \\ \leq \lambda_n(\phi_0(\theta_0) - f^* + \Phi) + D_n \to_{n\to\infty} D_\infty \end{gathered} \tag{24}$$

Proof.

$$\begin{gathered} \mathbb{E}_n f_n(\theta_n) \leq \\ \leq \phi_n^* + D_n \leq \\ \leq \phi_n(\theta_n) + D_n \leq \\ \leq (1-\lambda_n) f_n(\theta_n) + \lambda_n \tilde{\phi}_{0,n}(\theta_n) + D_n \end{gathered}$$

Remembering that $\tilde{\phi}_{0,n}(x) = \phi_0(x) - \phi_0(\theta_n) + \phi_0(\theta_0)$ we get

$$\mathbb{E}_n f_n(\theta_n) \leq (1-\lambda_n) f_n(\theta_n) + \lambda_n \phi_0(\theta_0) + \lambda_n \Phi + D_n$$

so

$$\mathbb{E}_n f_n(\theta_n) - f_n^* \leq \lambda_n(\phi_0(\theta_0) - f_n(\theta_n)) + \lambda_n \Phi + D_n$$

REFERENCES

[1] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate o(1/k2)," *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[2] B. Polyak, "O nekotorykh sposobakh uskorenia skhodimosti iteratsionnykh methodov," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

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

[4] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on International Conference on Machine Learning*, ser. ICML, 2013, pp. 1139–1147.

[5] G. E. Hinton, "A practical guide to training restricted boltzmann machines." in *Neural Networks: Tricks of the Trade (2nd ed.)*. Springer, 2012, pp. 599–619.

[6] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[7] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in minecraft," in *Proceedings of the 31th Conference on Artificial Intelligence*, 2017, pp. 1553–1561.

[8] S. W. Lee, M. O. Heo, J. Kim, J. Kim, and B. T. Zhang, "Dual memory architectures for fast deep learning of stream data via an online-incremental-transfer strategy," 2015.

[9] T. Ganegedara, L. Ott, and F. Ramos, "Online adaptation of deep architectures with reinforcement learning," in *Proceedings of the 22nd European Conference on Artificial Intelligence*, 2016, pp. 577–585.

[10] L. Guo, "Stability of recursive stochastic tracking algorithms," *SIAM Journal on Control and Optimization*, vol. 32, no. 5, pp. 1195–1225, 1994.

[11] L. Guo and L. Ljung, "Performance analysis of general tracking algorithms," *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1388–1402, 1995.

[12] V. Y. Katkovnik and V. E. Kheisin, "Dynamic stochastic approximation of polynomials drifts," *Automation and Remote Control*, vol. 40, no. 5, pp. 700–708, 1979.

[13] B. Delyon and A. Juditsky, "Asymptotical study of parameter tracking algorithms," *SIAM Journal on Control and Optimization*, vol. 33, no. 1, pp. 323–345, 1995.

[14] A. Benveniste, P. Priouret, and M. Métivier, *Adaptive Algorithms and Stochastic Approximations*. Springer Science & Business Media, 2012, vol. 22.

[15] E. Eweda and O. Macchi, "Tracking error bounds of adaptive nonstationary filtering," *Automatica*, vol. 21, no. 3, pp. 293–302, 1985.

[16] O. Granichin, L. Gurevich, and A. Vakhitov, "Discrete-time minimum tracking based on stochastic approximation algorithm with randomized differences," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 5763–5767.

[17] O. Granichin and N. Amelina, "Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1653–1658, 2015.

[18] H. J. Kushner and H. Huang, "Asymptotic properties of stochastic approximations with constant coefficients," *SIAM Journal on Control and Optimization*, vol. 19, no. 1, pp. 87–105, 1981.

[19] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media, 2003, vol. 35.

[20] V. S. Borkar, *Stochastic approximation*. Cambridge Books, 2008.

[21] A. Y. Popkov, "Gradient methods for nonstationary unconstrained optimization problems," *Automation and Remote Control*, vol. 66, no. 6, pp. 883–891, 2005.

[22] J. C. Spall. John Wiley & Sons, Inc., 2005, vol. 65.

[23] Y. E. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2013, vol. 87.

[24] A. T. Vakhitov, O. N. Granichin, and S. Sysoev, "A randomized stochastic optimization algorithm: Its estimation accuracy," *Automation and Remote Control*, vol. 67, no. 4, pp. 589–597, 2006.