EURO Journal on Computational Optimization 12 (2024) 100088

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

# A variable metric proximal stochastic gradient method: An application to classification problems

Pasquale Cascarano [a,1], Giorgia Franchini [c,*,1], Erich Kobler [d], Federica Porta [c,1], Andrea Sebastiani [b,c,1]

[a] Department of the Arts, University of Bologna, Bologna, Italy
[b] Department of Mathematics, University of Bologna, Bologna, Italy
[c] Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy
[d] Department of Neuroradiology, University Hospital Bonn, Bonn, Germany

ARTICLE INFO

*Keywords:*
Variable metric
Stochastic optimization
Classification problem
Deep learning

ABSTRACT

Due to the continued success of machine learning and deep learning in particular, supervised classification problems are ubiquitous in numerous scientific fields. Training these models typically involves the minimization of the empirical risk over large data sets along with a possibly non-differentiable regularization. In this paper, we introduce a stochastic gradient method for the considered classification problem. To control the variance of the objective's gradients, we use an automatic sample size selection along with a variable metric to precondition the stochastic gradient directions. Further, we utilize a non-monotone line search to automatize step size selection. Convergence results are provided for both convex and non-convex objective functions. Extensive numerical experiments verify that the suggested approach performs on par with state-of-the-art methods for training both statistical models for binary classification and artificial neural networks for multi-class image classification. The code is publicly available at https://github.com/koblererich/lisavm.

## 1. Introduction

Supervised classification problems arise in many real-life applications such as image recognition [1], web content filtering [2], medical diagnostics [3], analysis of genetic sequences [4] and biological systems [5,6], making them a challenging area of investigation. Both binary and multi-class classification problems involve the minimization of an objective function which can be formalized as the sum of cost functions whose number depends on the number of samples of given training set. Particularly, in this paper, we are interested in the following optimization problem

$$\min_{x\in\mathbb{R}^d} P(x) := \min_{x\in\mathbb{R}^d} F(x) + R(x) = \min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(x) + R(x), \tag{1}$$

where $F\colon \mathbb{R}^d \to \mathbb{R}$ is the so-called loss function and it computes the difference between the actual ground-truth and predicted values, $R\colon \mathbb{R}^d \to \mathbb{R}$ is a regularization term adding a priori information, $N$ represents the number of training samples and $d$ is the

* Corresponding author.
*E-mail addresses:* pasquale.cascarano2@unibo.it (P. Cascarano), giorgia.franchini@unimore.it (G. Franchini), kobler@uni-bonn.de (E. Kobler), federica.porta@unimore.it (F. Porta), andrea.sebastiani3@unibo.it (A. Sebastiani).
[1] INdAM-GNCS Research group, Roma, Italy.

https://doi.org/10.1016/j.ejco.2024.100088

Available online 15 April 2024

number of parameters. Each $f_i : \mathbb{R}^d \to \mathbb{R}$ denotes the loss function related to the $i$-th instance of the training set. We suppose that $F$ is continuously differentiable but possibly non-convex, while $R$ is convex but possibly non-differentiable. The class of proximal gradient algorithms [7,8] has been designed to solve optimization problems of this kind since, in their definition, they exploit the differentiability of $F$ and the convexity of $R$. However, in general, $N$ is very large making the computation of $F$ and its gradient prohibitively expensive. For this reason, proximal stochastic gradient schemes are typically exploited for optimization problems in classification applications. In more detail, the general iteration of the Proximal Stochastic Gradient (Prox-SG) method is

$$x^{(k+1)} = \text{prox}_{\alpha_k R}(x^{(k)} - \alpha_k g_{\mathcal{N}_k}(x^{(k)})), \tag{2}$$

where, given a sample $\mathcal{N}_k$ of size $N_k \ll N$ randomly and uniformly chosen from $\{1, \dots, N\}$,

$$g_{\mathcal{N}_k}(x) := \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} \nabla f_i(x)$$

is an unbiased estimator of the gradient of $F$ and $\alpha_k$ is a positive parameter typically called *learning rate* or *step size*. It is well known in the literature that determining proper values for both $\alpha_k$ and $N_k$ is a nontrivial task and it affects both the convergence properties and the numerical performance of the method itself. In particular, it is a common practice to decrease the step size $\alpha_k$ to train a neural network but it has been extensively proved that adaptively increasing sample size $N_k$ can achieve similar performances [9,10]. Moreover, the relationship between $\alpha_k$ and $N_k$ to guarantee an optimal trade-off between the parallelization benefits on bigger sample sizes and the generalization performances were studied in several works [11,12]. Other very popular approaches to face (1) are variance-reduced methods such as Proximal SVRG (Prox-SVRG) [13], Proximal SARAH (Prox-SARAH) [14]. Since these schemes need to periodically compute the full gradient of $F$ along the iterations, they are not employed in practical deep learning applications. For standard stochastic gradient methods to solve the non-regularized version of (1), variance reduction can be also achieved by either dynamic sampling strategies [15–18] or momentum-based techniques [19].

In [20], the authors suggested a proximal stochastic gradient method, called Prox-LISA, which is practically based on a monotone line search procedure to select the learning rate and exploits a dynamic increase of the sample size to compute the stochastic gradient. Despite promising properties, Prox-LISA suffers from some drawbacks: (i) the convergence behavior is typically not faster than that of Prox-SG; (ii) the learning rate could be reduced too fast along the iterations through the monotone line search; (iii) the theoretical properties on the objective function can be practically guaranteed in the convex case only; (iv) extensive numerical experiments on deep learning framework have not been performed.

*Contributions.* The main aim of this paper is to develop an improved version of Prox-LISA able to overcome the difficulties previously recalled. Below, we list the main ingredients that characterize this new version.

- A proper sequence of scaling matrices multiplying the stochastic directions will be considered in the updating step. The goal is to accelerate the performance of Prox-LISA by emulating a well-known behavior of deterministic proximal gradient methods. Indeed it is widely recognized that these schemes benefit from the presence of a variable metric underlying the iterates in terms of convergence speed [21–24]. We will specify which properties the sequence of scaling matrices must satisfy to guarantee convergence results and we will suggest how to practically define this sequence. It is worth mentioning that to practically define a proper sequence of scaling matrices is far from a straightforward task. Indeed, also in the deterministic framework, a predefined recipe to select suitable scaling matrices possibly does not exist or is strictly related to the application to be considered [21,22].
- The Prox-LISA scheme is based on a progressive increase of the sample size along the iterations to ensure a sufficient reduction of the variance of the stochastic directions. A refined dynamical technique to fix the sample size will be proposed. In contrast to the original criterion, the newly proposed scheme for updating the sample size allows also a decrease along the iterations. Moreover, the new strategy takes into account not only the current stochastic gradient but also the previous ones. The objective is to prevent an excessive amount of confidence from being placed on the current stochastic direction, especially in the early phase of the iterative process. We remark that the dynamical strategy for increasing the sample size followed in this work is different from those proposed in [15,17,18]. Indeed in these papers, the authors exploit the so-called *norm test* and its variants to monitor the variance reduction. These approaches can lead to a more significant increase in the sample size [25].
- Inspired by [26], a non-monotone line search to practically select the learning rate will be introduced. The non-monotonicity will be provided by a summable sequence which depends on the approximate width of the confidence interval built on the values of the sampled objective function. Hence, its definition automatically attempts to fit the features of the problem to be solved.

From the theoretical point of view, under proper assumptions on the variance of the current stochastic gradient, the decrease of the objective function in expectation, and the scaling matrices, the stationarity of the limit points of the sequence generated by the proposed scheme can be proven almost surely. If moreover, the objective function is convex, the whole sequence of the iterates converges to a solution almost surely. Finally, we show that if the learning rate is properly bounded then the condition on the decrease of the objective function in expectation is ensured even in the non-convex setting.

So far, the Prox-LISA method has been proven to be effective in training neural networks with just a few hidden layers [25]. In this paper, extensive numerical experiments show that the developed algorithm outperforms Prox-LISA and it is competitive with popular state-of-the-art methods for training both statistical models for binary classification and deep neural networks for multi-class image classification, even with large data sets. Notably, the proposed approach does not need manual tuning of the hyperparameters.

*Notations*

- Given $\mu \geq 1$, we denote by $\mathcal{M}_\mu$ the set of all symmetric positive definite matrices with all eigenvalues contained in the interval $\left[\frac{1}{\mu}, \mu\right]$. For any $D \in \mathcal{M}_\mu$, we have that $D^{-1}$ belongs to $\mathcal{M}_\mu$ and for any $x \in \mathbb{R}^d$

$$\frac{1}{\mu}\|x\|^2 \leq \|x\|_D^2 \leq \mu\|x\|^2. \tag{3}$$

- Let $D_1, D_2 \in \mathbb{R}^{d\times d}$ be symmetric and positive definite matrices. The notation $D_1 \succeq D_2$ indicates that $D_1 - D_2$ is a symmetric and positive semidefinite matrix or, equivalently, $x^T D_1 x \geq x^T D_2 x$ for any $x \in \mathbb{R}^d$.
- Let $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a closed, convex, proper (CCP) function, and let $\lambda$ be a positive scalar. Given $x \in \mathbb{R}^n$, the proximal operator associated to the function $\lambda f$ in the metric induced by a symmetric and positive definite matrix $D$ is defined as

$$\text{prox}_{\lambda f}^D(x) = \underset{y\in\mathbb{R}^n}{\text{argmin}}\ f(y) + \frac{1}{2\lambda}\|y - x\|_D^2.$$

## 2. The method and its convergence analysis

To solve the minimization problem (1), we consider the following class of variable metric stochastic gradient methods

$$x^{(k+1)} = \text{prox}_{\alpha_k R}^{D_k}\left(x^{(k)} - \alpha_k D_k^{-1} g_{\mathcal{N}_k}(x^{(k)})\right), \tag{4}$$

where the learning rate $\alpha_k$ is a positive parameter, and $D_k$ is a symmetric and positive definite matrix. Hereafter, we denote by $e(x)$ the residual between the gradient of $F$ and its stochastic approximation, namely

$$e(x) := g_{\mathcal{N}_k}(x) - \nabla F(x).$$

Let $\mathcal{F}_k$ be the $\sigma$-algebra generated by $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$, we suppose that the gradient estimator $g_{\mathcal{N}_k}$ is unbiased, namely

$$\mathbb{E}(e(x^{(k)}) \mid \mathcal{F}_k) = 0.$$

The following convergence analysis generalizes the one proposed in [20] to the presence of the variable metric induced by the sequence of the scaling matrices $\{D_k\}$. We highlight the assumptions which have to be imposed on $\{D_k\}$ and how they contribute to the convergence analysis. We have omitted the proofs of Theorems 1 and 2 that can be readily derived from the analysis in the non-scaled framework [20]. Instead, we provide a comprehensive explanation for all the arguments of the proofs that are not straightforward in the variable metric setting.
Firstly, we observe that, given the function

$$h^{(k)}(z) := \nabla F(x^{(k)})^T(z - x^{(k)}) + \frac{1}{2\alpha_k}\|z - x^{(k)}\|_{D_k}^2 + R(z) - R(x^{(k)}), \tag{5}$$

it holds that from the convexity of $R(z)$

$$\begin{aligned} p(x) &:= \text{prox}_{\alpha_k R}^{D_k}\left(x^{(k)} - \alpha_k D_k^{-1}\nabla F(x^{(k)})\right) = \text{argmin}_{z\in\mathbb{R}^d}\ \frac{1}{2\alpha_k}\|z - x^{(k)} + \alpha_k D_k^{-1}\nabla F(x^{(k)})\|_{D_k}^2 + R(z) = \\ &= \text{argmin}_{z\in\mathbb{R}^d}\ h^{(k)}(z), \end{aligned} \tag{6}$$

while the $(k+1)$-th iterate of (4) can be written as

$$x^{(k+1)} = \text{argmin}_{z\in\mathbb{R}^d}\ h^{(k)}(z) + e(x^{(k)})^T(z - x^{(k)}). \tag{7}$$

As a consequence, $\forall\ z \in \mathbb{R}^d$,

$$h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \leq h^{(k)}(z) + e(x^{(k)})^T(z - x^{(k)}),$$

and hence, by setting $z = x^{(k)}$, (5) yields

$$h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \leq 0. \tag{8}$$

Before introducing the convergence results, we detail the assumptions on the objective function and some useful Lemmas. In particular, we assume that the functions involved in the problem (1) have the following properties.

(i) $R : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is a proper, convex, and lower semicontinuous function, with a non-empty and closed domain.
(ii) $F : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is a continuously differentiable function on an open subset $Y$ of $\mathbb{R}^d$ containing $dom(R)$.
(iii) $P$ is bounded from below on $dom(R) \cap dom(F)$ and $x^* \in X^* := \text{argmin}_x P(x) \neq \emptyset$.
(iv) All the $f_i$ have Lipschitz-continuous gradients with Lipschitz constant $L$.

As a consequence of assumption (iv), the gradient estimate $g_{\mathcal{N}_k}(x)$ and $\nabla F$ are both Lipschitz continuous with Lipschitz parameter $L$.

Lemma 1 states a standard result from stochastic analysis on supermartingale convergence. Lemma 2 recollects some useful results on both the proximal operator defined with respect to a variable metric and the function $h^{(k)}$ defined in (5).

**Lemma 1.** *[27, Lemma 11] Let $\nu_k$, $u_k$, $\gamma_k$, $\beta_k$ be non-negative random variables and let*

$$\mathbb{E}(\nu_{k+1} \mid \mathcal{F}_k) \leq (1+\gamma_k)\nu_k - u_k + \beta_k \qquad a.s.$$
$$\sum_{k=0}^{\infty} \gamma_k < \infty \quad a.s., \qquad \sum_{k=0}^{\infty} \beta_k < \infty \quad a.s.,$$

*where $\mathbb{E}(\nu_{k+1} \mid \mathcal{F}_k)$ denotes the conditional expectation for the given $\nu_0, \dots, \nu_k$, $u_0, \dots, u_k$, $\gamma_0, \dots, \gamma_k$, $\beta_0, \dots, \beta_k$. Then*

$$\nu_k \longrightarrow \nu \quad a.s, \qquad \sum_{k=0}^{\infty} u_k < \infty \quad a.s,$$

*where $\nu \geq 0$ is some random variable.*

**Lemma 2.** *Let $\alpha_k \in [\underline{\alpha}, \overline{\alpha}]$, $\underline{\alpha} > 0$ and $D_k \in \mathcal{M}_\mu$, $x^{(k)} \in dom(P)$. The following statements hold true.*

a. $\hat{y} = \operatorname{prox}^{D_k}_{\alpha_k R}(x^{(k)} - \alpha_k D_k^{-1} u)$ *if and only if* $\frac{1}{\alpha_k} D_k(x^{(k)} - \hat{y}) - u = w$, $w \in \partial R(\hat{y})$.
b. *The function $h^{(k)}$ is strongly convex with modulus of convexity* $\frac{1}{\overline{\alpha}\mu}$.
c. $h^{(k)}(x^{(k)}) = 0$.
d. $h^{(k)}(p(x^{(k)})) \leq 0$ *and* $h^{(k)}(p(x^{(k)})) = 0$ *if and only if* $p(x^{(k)}) = x^{(k)}$.
e. $x^{(k)}$ *is a stationary point for problem (1) if and only if* $x^{(k)} = p(x^{(k)})$.
f. $x^{(k)}$ *is a stationary point for problem (1) if and only if* $h^{(k)}(p(x^{(k)})) = 0$.

**Proof.** For the proof of item *a.*, *c.*, *d.*, *e.* and *f.* we refer the reader to [21,28]. As for item *b.*, just observe that

$$\left(\frac{1}{\alpha_k} D_k(z - x^{(k)}) - \frac{1}{\alpha_k} D_k(y - x^{(k)})\right)^T (z - y) = \frac{1}{\alpha_k}\|z-y\|^2_{D_k} \geq \frac{1}{\overline{\alpha}}\|z-y\|^2_{D_k} \geq \frac{1}{\overline{\alpha}\mu}\|z-y\|^2. \quad \square$$

Lemma 3 generalizes [20, Lemma 2] to the variable metric framework. Additionally, the overall arguments of the proof of Lemma 3 are somewhat simplified compared to those of the proof of [20, Lemma 2].

**Lemma 3.** *Let us consider the sequence $\{x^{(k)}\}$ generated by the iteration (4). If $\alpha_k > 0$ and $D_k$ is a symmetric and positive definite matrix $\forall k$, the following inequality holds:*

$$h^{(k)}(x^{(k+1)}) - h(p(x^{(k)})) \leq \frac{\alpha_k}{2}\|e(x^{(k)})\|^2_{D_k^{-1}}. \tag{9}$$

**Proof.**

$$\begin{aligned}
h^{(k)}(x^{(k+1)}) - h^{(k)}(p(x^{(k)})) &= \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \nabla F(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + R(x^{(k+1)}) - R(x^{(k)}) + \\
&- \frac{1}{2\alpha_k}\|p(x^{(k)}) - x^{(k)}\|^2_{D_k} - \nabla F(x^{(k)})^T(p(x^{(k)}) - x^{(k)}) - R(p(x^{(k)})) + R(x^{(k)}) = \\
&= \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \nabla F(x^{(k)})^T(x^{(k+1)} - p(x^{(k)})) - \frac{1}{2\alpha_k}\|p(x^{(k)}) - x^{(k)}\|^2_{D_k} + \\
&+ R(x^{(k+1)}) - R(p(x^{(k)})) \\
&\leq \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \nabla F(x^{(k)})^T(x^{(k+1)} - p(x^{(k)})) - \frac{1}{2\alpha_k}\|p(x^{(k)}) - x^{(k)}\|^2_{D_k} + \\
&+ \frac{1}{\alpha_k}(x^{(k)} - x^{(k+1)})^T D_k(x^{(k+1)} - p(x^{(k)})) - (\nabla F(x^{(k)}) + e(x^{(k)}))^T(x^{(k+1)} - p(x^{(k)})) = \\
&= \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)} + p(x^{(k)}) - p(x^{(k)})\|^2_{D_k} - \frac{1}{2\alpha_k}\|p(x^{(k)}) - x^{(k)}\|^2_{D_k} + \\
&+ \frac{1}{\alpha_k}(x^{(k)} - x^{(k+1)})^T D_k(x^{(k+1)} - p(x^{(k)})) - e(x^{(k)})^T(x^{(k+1)} - p(x^{(k)})) = \\
&= \frac{1}{2\alpha_k}\|x^{(k+1)} - p(x^{(k)})\|^2_{D_k} + \frac{1}{\alpha_k}(x^{(k+1)} - p(x^{(k)}))^T D_k(p(x^{(k)}) - x^{(k)}) +
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{\alpha_k}(x^{(k)} - x^{(k+1)})^T D_k(x^{(k+1)} - p(x^{(k)})) + e(x^{(k)})^T(x^{(k+1)} - p(x^{(k)})) = \\
&= -\frac{1}{2\alpha_k}\|x^{(k+1)} - p(x^{(k)})\|^2_{D_k} + e(x^{(k)})^T(x^{(k+1)} - p(x^{(k)})) = \\
&= -\frac{1}{2\alpha_k}\|x^{(k+1)} - p(x^{(k)}) - \alpha_k D_k^{-1} e(x^{(k)})\|^2_{D_k} + \frac{1}{2\alpha_k}\|\alpha_k D_k^{-1} e(x^{(k)})\|^2_{D_k} \\
&\leq \frac{1}{2\alpha_k}\|\alpha_k D_k^{-1} e(x^{(k)})\|^2_{D_k} = \frac{\alpha_k}{2}\|D_k^{-1} e(x^{(k)})\|^2_{D_k} = \frac{\alpha_k}{2}\|e^{(k)}\|^2_{D_k^{-1}}.
\end{aligned}
$$

The first inequality follows from the convexity of $R$ and the fact that $\frac{1}{\alpha_k} D_k(x^{(k)} - x^{(k+1)}) - (\nabla F(x^{(k)}) + e(x^{(k)})) \in \partial R(x^{(k+1)})$ (Lemma 2, part *a.*). The second and third equality can be derived through fundamental vector calculations. Finally, the last inequality follows from the non-positivity of the first term. □

Theorem 1 introduces a crucial condition on the decrease of the objective function in expectation needed for the convergence results. The feasibility of this condition in the practice will be discussed in Remark 1.

**Theorem 1.** *Let $\{x^{(k)}\}$ be the sequence generated by the method (4) where $\alpha_k \in [\underline{\alpha}, \overline{\alpha}]$, $\underline{\alpha} > 0$ and $D_k$ is symmetric and positive definite matrices $\forall k$. Let $0 < \gamma \leq 1$ and $\{\eta_k\}_{k\in\mathbb{N}}$ be a sequence of non-negative random variables such that $\sum_{k=0}^{\infty} \eta_k < \infty$ a.s. If, for any $x^{(0)} \in dom(P)$,*

$$
\mathbb{E}(P(x^{(k+1)}) \mid \mathcal{F}_k) \leq \mathbb{E}(P(x^{(k)}) \mid \mathcal{F}_k) + \gamma\mathbb{E}(h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \mid \mathcal{F}_k) + \eta_k, \tag{10}
$$

*then, $P(x^{(k)}) - P^* \longrightarrow \overline{P}$ a.s., where $\overline{P} \geq 0$ is some random variable and $P^*$ is such that $P(x) \geq P^*$, for $x \in dom(P)$. Furthermore, the following assertions hold:*

i) $\sum_{k=0}^{\infty} \mathbb{E}\left(-h^{(k)}(x^{(k+1)}) - e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \mid \mathcal{F}_k\right) < \infty$ *a.s.,*
ii) $h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) \to 0$ *a.s.*

**Proof.** The proof of this theorem follows the one of [20, Theorem 1]. □

**Remark 1.** It is worth to detail when condition (10) can practically be met. In their work [20], the authors show that condition (10) can be satisfied provided that the objective function is convex. Here we take a step further and we clarify how to guarantee condition (10) in the non convex setting. Indeed, since $\nabla F$ is $L$-Lipschitz continuous and $D_k \in \mathcal{M}_\mu$, we have

$$
\begin{aligned}
F(x^{(k+1)}) + R(x^{(k+1)}) &\leq F(x^{(k)}) + R(x^{(k)}) + \nabla F(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{L}{2}\|x^{(k+1)} - x^{(k)}\|^2 + R(x^{(k+1)}) - R(x^{(k)}) \\
&\quad + \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} - \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} \\
&\leq F(x^{(k)}) + R(x^{(k)}) + \nabla F(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{L\mu}{2}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + R(x^{(k+1)}) - R(x^{(k)}) \\
&\quad + \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k} - \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|^2_{D_k}
\end{aligned} \tag{11}
$$

where the first inequality follows from the Lipschitz-continuity of $\nabla F$ and the second is a consequence of $D_k \in \mathcal{M}_\mu$. Consequently, given the definition (5) we can write

$$
P(x^{(k+1)}) \leq P(x^{(k)}) + h^{(k)}(x^{(k+1)}) + \frac{1}{2}\left(L\mu - \frac{1}{\alpha_k}\right)\|x^{(k+1)} - x^{(k)}\|^2_{D_k}.
$$

By adding and subtracting $e(x^{(k)})^T(x^{(k+1)} - x^{(k)})$, we obtain

$$
\begin{aligned}
P(x^{(k+1)}) &\leq P(x^{(k)}) + h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2}\left(L\mu - \frac{1}{\alpha_k}\right)\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \\
&\quad - \overline{\alpha} e(x^{(k)})^T \frac{x^{(k+1)} - x^{(k)}}{\overline{\alpha}} \\
&\leq P(x^{(k)}) + h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2}\left(L\mu - \frac{1}{\alpha_k} + \frac{1}{\overline{\alpha}}\right)\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \\
&\quad + \frac{\overline{\alpha}}{2}\|e(x^{(k)})\|^2_{D_k} \\
&\leq P(x^{(k)}) + h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2}\left(L\mu - \frac{1}{\alpha_k} + \frac{1}{\overline{\alpha}}\right)\|x^{(k+1)} - x^{(k)}\|^2_{D_k} +
\end{aligned}
$$

$$+\frac{\overline{\alpha}\mu}{2}\|e(x^{(k)})\|^2 \tag{12}$$

where the inequality $-\overline{\alpha}e(x^{(k)})^T\frac{x^{(k+1)}-x^{(k)}}{\overline{\alpha}} \leq \frac{1}{2\overline{\alpha}}\|x^{(k+1)}-x^{(k)}\|^2_{D_k}+\frac{\overline{\alpha}}{2}\|e(x^{(k)})\|^2_{D_k}$ has been used. Thus, by considering the conditional expectation in both members, it is possible to find a value of $\alpha_k < \overline{\alpha} < \frac{1}{L\mu}$ such that $L\mu - \frac{1}{\alpha_k}+\frac{1}{\overline{\alpha}}<0$ (for example $\alpha_k=\frac{\overline{\alpha}}{2}$) and we can write

$$\mathbb{E}(P(x^{(k+1)})|\mathcal{F}_k) \leq \mathbb{E}(P(x^{(k)})+h^{(k)}(x^{(k+1)};x^{(k)})+e(x^{(k)})^T(x^{(k+1)}-x^{(k)})|\mathcal{F}_k)+\frac{\overline{\alpha}\mu}{2}\mathbb{E}(\|e(x^{(k)})\|^2|\mathcal{F}_k).$$

Hence, if $\overline{\alpha}<\frac{1}{L\mu}$ and $\mathbb{E}(\|e(x^{(k)})\|^2|\mathcal{F}_k)\leq\varepsilon_k$ where $\sum_k\varepsilon_k<\infty$, then condition (10) is satisfied.

Theorem 2 state the conditions which ensure that any limit point of the sequence $\{x^{(k)}\}$ generated by (4) is a stationary point for (1) almost surely.

**Theorem 2.** *Let $\{x^{(k)}\}$ be the sequence generated by the method (4) with $\mathbb{E}(\|e(x^{(k)})\|^2 \mid \mathcal{F}_k)\leq\varepsilon_k$ where $\{\varepsilon_k\}$ is a non-negative sequence such that $\lim_{k\to\infty}\varepsilon_k=0$, $\alpha_k\in[\underline{\alpha},\overline{\alpha}],\underline{\alpha}>0$ and $D_k\in\mathcal{M}_\mu$. Moreover, suppose that the condition (10) is satisfied for any $x^{(0)}\in dom(P)$. Then any limit point of the sequence $\{x^{(k)}\}$ is stationary for problem (1) a.s.*

**Proof.** The proof of this theorem directly follows [20, Theorem 2] by accounting for the scaling matrix $D_k$ and the bounds on its eigenvalues. □

The last theorem of this section shows that if the objective function is convex, then the sequence $\{x^{(k)}\}$ obtained by (4) almost surely converges to a solution of problem (1).

**Theorem 3.** *Let $\{x^{(k)}\}$ be the sequence generated by the method (4) with $\mathbb{E}(\|e(x^{(k)})\|^2 \mid \mathcal{F}_k)\leq\varepsilon_k$ where $\{\varepsilon_k\}$ is a deterministic non-negative and non-increasing sequence such that $\sum_k\sqrt{\varepsilon_k}<+\infty$, $\alpha_k\in[\underline{\alpha},\overline{\alpha}],\underline{\alpha}>0$ and $D_k\in\mathcal{M}_\mu$, $\mu\geq1$. Moreover, suppose that*

$$D_{k+1}\preceq(1+\zeta_k)D_k \tag{13}$$

*where the deterministic sequence $\{\zeta_k\}_{k\in\mathbb{N}}$ satisfies the following conditions*

$$\{\zeta_k\}_{k\in\mathbb{N}}\subset\mathbb{R}_{\geq0},\quad \sum_{k=0}^{+\infty}\zeta_k<+\infty. \tag{14}$$

*Finally, suppose that condition (10) is satisfied for any $x^{(0)}\in dom(P)$ and the function $F$ is convex. Then the sequence $\{x^{(k)}\}$ converges to a solution of problem (1) a.s.*

**Proof.** Let $x^*\in X^*$. Since $\frac{1}{\alpha_k}D_k(x^{(k)}-x^{(k+1)})-g_{\mathcal{N}_k}(x^{(k)})\in\partial R(x^{(k+1)})$, it holds that

$$R(y)\geq R(x^{(k+1)})+\frac{1}{\alpha_k}\left(x^{(k)}-x^{(k+1)}-\alpha_k D_k^{-1}g_{\mathcal{N}_k}(x^{(k)})\right)^T D_k\left(y-x^{(k+1)}\right),\quad\forall y\in\mathbb{R}^d.$$

It follows that, $\forall y\in\mathbb{R}^d$,

$$(x^{(k+1)}-x^{(k)})^T D_k(y-x^{(k+1)})\geq\alpha_k\left(R(x^{(k+1)})-R(y)+g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k+1)}-y)\right). \tag{15}$$

For $y=x^*$ the previous inequality gives

$$(x^{(k+1)}-x^{(k)})^T D_k(x^*-x^{(k)}+x^{(k)}-x^{(k+1)})\geq\alpha_k\left(R(x^{(k+1)})-R(x^*)+g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k+1)}-x^{(k)}+x^{(k)}-x^*)\right).$$

As a consequence, we obtain the following relations:

$$\begin{aligned}
(x^{(k+1)}-x^{(k)})^T D_k(x^*-x^{(k)}) &\geq \alpha_k\left(R(x^{(k+1)})-R(x^*)+g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k)}-x^*)\right)+\|x^{(k+1)}-x^{(k)}\|^2_{D_k}\\
&\quad+\alpha_k g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k+1)}-x^{(k)})\\
&\geq \alpha_k\left(R(x^{(k+1)})-R(x^*)+F(x^{(k)})-F(x^*)\right)+\alpha_k e(x^{(k)})^T(x^{(k)}-x^*)+\\
&\quad+\|x^{(k+1)}-x^{(k)}\|^2_{D_k}+\alpha_k(\nabla F(x^{(k)})+e(x^{(k)}))^T(x^{(k+1)}-x^{(k)})\\
&= \alpha_k\left(R(x^{(k+1)})-R(x^{(k)})+P(x^{(k)})-P(x^*)\right)+\|x^{(k+1)}-x^{(k)}\|^2_{D_k}+\\
&\quad+\alpha_k e(x^{(k)})^T(x^{(k)}-x^*)+\alpha_k(\nabla F(x^{(k)})+e(x^{(k)}))^T(x^{(k+1)}-x^{(k)})
\end{aligned}$$

$$\geq \alpha_k \left( R(x^{(k+1)}) - R(x^{(k)}) \right) + \|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \alpha_k e(x^{(k)})^T (x^{(k)} - x^*) + \\ + \alpha_k (\nabla F(x^{(k)}) + e(x^{(k)}))^T (x^{(k+1)} - x^{(k)}), \tag{16}$$

where the second inequality follows from the convexity of $F$ and the last inequality follows from the fact that $P(x^{(k)}) - P(x^*) \geq 0$. From a basic property of the Euclidean norm[2] and (16) we can write

$$\begin{aligned}
\|x^{(k+1)} - x^*\|^2_{D_k} &= \|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \|x^{(k)} - x^*\|^2_{D_k} - 2(x^{(k+1)} - x^{(k)})^T D_k (x^* - x^{(k)}) \\
&\overset{(16)}{\leq} \|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \|x^{(k)} - x^*\|^2_{D_k} - 2\alpha_k \left( R(x^{(k+1)}) - R(x^{(k)}) \right) - 2\|x^{(k+1)} - x^{(k)}\|^2_{D_k} + \\
&\quad - 2\alpha_k e(x^{(k)})^T (x^{(k)} - x^*) - 2\alpha_k (\nabla F(x^{(k)}) + e(x^{(k)}))^T (x^{(k+1)} - x^{(k)}) \\
&= \|x^{(k)} - x^*\|^2_{D_k} - 2\alpha_k e(x^{(k)})^T (x^{(k)} - x^*) - 2\alpha_k e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) + \\
&\quad - 2\alpha_k \left( R(x^{(k+1)}) - R(x^{(k)}) + \nabla F(x^{(k)})^T (x^{(k+1)} - x^{(k)}) + \frac{1}{2\alpha_k} \|x^{(k+1)} - x^{(k)}\|^2_{D_k} \right) \\
&= \|x^{(k)} - x^*\|^2_{D_k} - 2\alpha_k \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) - 2\alpha_k e(x^{(k)})^T (x^{(k)} - x^*) \\
&\leq \|x^{(k)} - x^*\|^2_{D_k} - 2\overline{\alpha} \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) - 2\alpha_k e(x^{(k)})^T (x^{(k)} - x^*).
\end{aligned}$$

This inequality combined with assumption (13) allows to state that

$$\begin{aligned}
\|x^{(k+1)} - x^*\|^2_{D_k} &\leq (1 + \zeta_{k-1}) \|x^{(k)} - x^*\|^2_{D_{k-1}} - 2\overline{\alpha} \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) - 2\alpha_k e(x^{(k)})^T (x^{(k)} - x^*) \\
&\leq (1 + \zeta_{k-1}) \|x^{(k)} - x^*\|^2_{D_{k-1}} - 2\overline{\alpha} \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) + \overline{\alpha} \frac{\|e(x^{(k)})\|^2}{\sqrt{\varepsilon_k}} + \\
&\quad + \overline{\alpha} \sqrt{\varepsilon_k} \|x^{(k)} - x^*\|^2 \\
&\leq (1 + \zeta_{k-1}) \|x^{(k)} - x^*\|^2_{D_{k-1}} - 2\overline{\alpha} \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) + \overline{\alpha} \frac{\|e(x^{(k)})\|^2}{\sqrt{\varepsilon_k}} + \\
&\quad + \overline{\alpha} \mu \sqrt{\varepsilon_k} \|x^{(k)} - x^*\|^2_{D_{k-1}} \\
&= (1 + \zeta_{k-1} + \overline{\alpha} \mu \sqrt{\varepsilon_k}) \|x^{(k)} - x^*\|^2_{D_{k-1}} - 2\overline{\alpha} \left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) \right) + \overline{\alpha} \frac{\|e(x^{(k)})\|^2}{\sqrt{\varepsilon_k}},
\end{aligned}$$

where the second inequality follows from the fact that for any $a, b \in \mathbb{R}^d$ and $\varepsilon > 0$ it holds that $-2a^T b = \|a/\varepsilon\|^2 + \|\varepsilon b\|^2 - \|a/\varepsilon + \varepsilon b\|^2 \leq \|a/\varepsilon\|^2 + \|\varepsilon b\|^2$. Taking the conditional expectation with respect to the $\sigma$-algebra $\mathcal{F}_k$ and recalling that the sequences $\{\varepsilon_k\}$ and $\{\zeta_k\}$ are a priori fixed, we obtain

$$\mathbb{E}\left( \|x^{(k+1)} - x^*\|^2_{D_k} | \mathcal{F}_k \right) \leq (1 + \zeta_{k-1} + \overline{\alpha} \mu \sqrt{\varepsilon_k}) \|x^{(k)} - x^*\|^2_{D_{k-1}} - 2\overline{\alpha} \mathbb{E}\left( h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T (x^{(k+1)} - x^{(k)}) | \mathcal{F}_k \right) + \\ + \overline{\alpha} \sqrt{\varepsilon_k}.$$

By combining this last inequality and part *i)* of Theorem 1 together with Lemma 1 where $\gamma_k = \zeta_{k-1} + \overline{\alpha} \mu \sqrt{\varepsilon_k}$, we can state that the sequence $\{\|x^{(k)} - x^*\|_{D_{k-1}}\}_{k \in \mathbb{N}}$ converges a.s. Since $D_{k-1} \in \mathcal{M}_\mu$, $\{\|x^{(k)} - x^*\|\}_{k \in \mathbb{N}}$ converges a.s. too. The proof can be concluded as the one of Theorem 3 in [20]. □

Condition (13) states that the sequence $\{D_k\}_{k \in \mathbb{N}}$ asymptotically approaches a constant matrix [29, Lemma 2.3]. In Section 3 we discuss how to satisfy condition (13) in practice.

**Remark 2.** We conclude this section by noting that under the hypotheses of Theorem 3, a convergence rate result analogous to that of [20, Theorem 4] also holds for the sequence $\{x^{(k)}\}$ generated by the method (4). Starting from inequality (16), the proof follows as in [30, Theorem 4].

## 3. Practical implementation

In this section, we detail how to select the hyperparameters defining iteration (4), namely the sequences $\{\alpha_k\}, \{D_k\}$ and $\{\mathcal{N}_k\}$, to practically realize the theoretical conditions stated in Theorems 1-3. Algorithm 1 lists the main steps of the proposed method.

[2] $\|a - b\|^2_D + \|b - c\|^2_D - \|a - c\|^2_D = 2(a - b)^T D(c - b), \quad \forall a, b, c \in \mathbb{R}^d$.

**Selection of the sample $\mathcal{N}_k$ (Step 1.)** To guarantee that the theoretical assumption (10) holds in the practice, the condition

$$\mathbb{E}(\|e(x^{(k)})\|^2 \mid \mathcal{F}_k) \leq \varepsilon_k, \quad \sum_{k=0}^{+\infty} \varepsilon_k < +\infty \tag{17}$$

must be satisfied. However, inequality (17) involves the computation of $\nabla F(x^{(k)})$. As a consequence, we need to consider an approximate criterion that exploits the sample's variance. In more detail, since for an arbitrary $i \in \mathcal{N}_k$ [31, pg. 183]

$$\mathbb{E}\left(\frac{1}{2}\|e(x^{(k)})\|_2^2 \mid \mathcal{F}_k\right) \leq \frac{1}{2N_k}\mathbb{E}\left(\|\nabla f_i(x^{(k)}) - \nabla F(x^{(k)})\|_2^2 \mid \mathcal{F}_k\right), \tag{18}$$

as a practical counterpart of (17), one could consider

$$V_{\mathcal{N}_k}(x^{(k)}) := \frac{1}{2N_k(N_k-1)} \sum_{i \in \mathcal{N}_k} \|\nabla f_i(x^{(k)}) - g_{\mathcal{N}_k}(x^{(k)})\|_2^2 \leq \gamma_1 \varepsilon_k, \tag{19}$$

where the right-hand side of (18) has been approximated by the sample's variance, borrowing similar strategies as exploited in [16,20]. As a result, the variance is controlled by a vanishing nonnegative sequence $\gamma_1\varepsilon_k$, where the positive scalar $\gamma_1$ needs to be adapted to the problem at hand to control the variance, especially at the beginning of the iterative process. In this work, we instead control the variance by the upper bound

$$\overline{V}^{(k)} := \min\left(\gamma_1\varepsilon_k, \frac{V^{(k-1)}}{1-\beta_1^{k-1}} + \gamma_2\sqrt{\frac{V_{\text{var}}^{(k-1)}}{1-\beta_2^{k-1}}}\right) \tag{20}$$

using $\beta_1, \beta_2 \in (0,1)$, positive constants $\gamma_1, \gamma_2$, any nonnegative sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$ such that $\sum_k \varepsilon_k < +\infty$, and the running statistics of the sample variance from the previous iteration

$$\begin{aligned} V^{(k-1)} &= \beta_1 V^{(k-2)} + (1-\beta_1) V_{\mathcal{N}_{k-1}}(x^{(k-1)}) \\ V_{\text{var}}^{(k-1)} &= \beta_2 V_{\text{var}}^{(k-2)} + (1-\beta_2)\left(V_{\mathcal{N}_{k-1}}(x^{(k-1)}) - V^{(k-1)}\right)^2. \end{aligned}$$

As a result, the upper variance bound $\overline{V}^{(k)}$ accounts for outliers along the iterative process due to the running statistics. In addition, it vanishes in the limit and thereby guarantees that (17) holds in the limit. Whenever

$$V_{\mathcal{N}_k}(x^{(k)}) \leq \overline{V}^{(k)} \tag{21}$$

is not satisfied, the sample size $N_k$ is increased as outlined in Step 1 of Algorithm 1. Note that we use an upper bound for the sample size $N_k$ in practice to account for limited hardware resources. Additionally, it is worth mentioning that our algorithm provides the flexibility to reduce the sample size. In Step 4, we decrease the attempt value for the next sample size by a factor of $\delta_2$ compared to the current size. In Step 1, if necessary, the sample size can be increased to ensure that inequality (21) is satisfied at each iteration.

**Selection of the scaling matrix $D_k$ (Step 2).** According to the hypotheses of Theorem 2 the sequence of the scaling matrices $\{D_k\}$ must fulfill the following condition

$$\{D_k\} \subseteq \mathcal{M}_\mu, \quad \mu \geq 1. \tag{22}$$

In practice, given $\mu > 1$, a possibility to realize a sequence $\{D_k\}$ satisfying condition (22) is to define the scaling matrix $D_k$ as the diagonal matrix

$$D_k = \text{diag}\left(\min\left(\mu, \max\left(d^{(k)}, \frac{1}{\mu}\right)\right)\right) \tag{23}$$

where $d^{(k)}$ is a proper vector. Inspired by the preconditioner employed in [32], we fix

$$d^{(k)} = \sqrt{\frac{g_{\text{var}}^{(k)}}{1-\beta_2^k}} + \epsilon \tag{24}$$

for

$$\begin{aligned} g^{(k)} &= \beta_1 g^{(k-1)} + (1-\beta_1) g_{\mathcal{N}_k}(x^{(k)}) \\ g_{\text{var}}^{(k)} &= \beta_2 g_{\text{var}}^{(k-1)} + (1-\beta_2)(g_{\mathcal{N}_k}(x^{(k)}) - g^{(k)})^2 + \epsilon, \end{aligned}$$

where $\beta_1, \beta_2 \in (0,1)$, $\epsilon > 0$, $g^{(0)}$ and $g_{\text{var}}^{(0)}$ are null vectors and the vector squaring is intended element-wise.
In order to ensure stronger convergence results (Theorem 3), the sequence of the scaling matrices $\{D_k\}$ must satisfy an additional assumption:

$$D_{k+1} \preceq (1+\zeta_k)D_k, \quad \{\zeta_k\} \subset \mathbb{R}_{\geq 0}, \quad \sum_{k=0}^{+\infty} \zeta_k < +\infty.$$

A possibility to fulfill this condition (see [33]) is to impose

$$\{D_k\} \subseteq \mathcal{M}_{\mu_k}, \quad \text{where} \quad \mu_k^2 = 1+\nu_k, \quad \{\nu_k\} \subset \mathbb{R}_{\geq 0}, \quad \sum_{k=0}^{+\infty} \nu_k < +\infty. \tag{25}$$

As a consequence, $D_k$ can be defined as in (23)-(24), but instead of $\mu$ and $1/\mu$, the diagonal entries must be bounded by $\mu_k$ and $1/\mu_k$.

**Selection of the learning rate $\alpha_k$ (Step 3.)** In order to ensure the validity of condition (10) we need to act on the learning rate $\alpha_k$, as shown in Remark 1. In particular, a proper bound on the learning rate should be imposed: $\overline{\alpha} < \frac{1}{L\mu}$. Since the Lipschitz constant $L$ is not always known we suggest estimating it employing an adaptive procedure. Firstly we recall that, given the sub-sampled function

$$f_{\mathcal{N}_k}(x) = \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} f_i(x) \tag{26}$$

and since $g_{\mathcal{N}_k}$ is $L$-Lipschitz continuous, the following inequality

$$f_{\mathcal{N}_k}(x^{(k+1)}) \leq f_{\mathcal{N}_k}(x^{(k)}) + g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2\alpha}\|x^{(k+1)} - x^{(k)}\|_{D_k}^2 \tag{27}$$

holds for all $\alpha < \frac{1}{L\mu}$ [33, Lemma 6]. As a consequence, we consider (27) to realize a line search strategy to achieve the bound for the learning rate sequence practically. A similar approach has been also exploited in [15,20,34] in the non-scaled framework. However, differently from [15,20,34], the line search proposed in this paper exploits a relaxed version of inequality (27). Indeed we force $\alpha_k$ to satisfy the following inequality

$$f_{\mathcal{N}_k}(x^{(k+1)}) \leq f_{\mathcal{N}_k}(x^{(k)}) + g_{\mathcal{N}_k}(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \frac{1}{2\alpha_k}\|x^{(k+1)} - x^{(k)}\|_{D_k}^2 + \tau_k \tag{28}$$

where $\{\tau_k\}$ is a non-negative summable sequence. In view of (27), it is always possible to find $\alpha_k$ such that condition (28) is satisfied. Accordingly, the line search at Step 3 of Algorithm 1 is well defined. Hereafter we clarify why the presence of the sequence $\{\tau_k\}$ can be seen as a parameter inducing a non-monotone behavior to the line search. By adding $R(x^{(k+1)}) - R(x^{(k)})$ to both sides of inequality (28) and recalling that $g_{\mathcal{N}_k}(x^{(k)}) = \nabla F(x^{(k)}) + e(x^{(k)})$, it holds that

$$f_{\mathcal{N}_k}(x^{(k+1)}) + R(x^{(k+1)}) \leq f_{\mathcal{N}_k}(x^{(k)}) + R(x^{(k)}) + h^{(k)}(x^{(k+1)}) + e(x^{(k)})^T(x^{(k+1)} - x^{(k)}) + \tau_k. \tag{29}$$

Thanks to (8), we can conclude that inequality (28) is equivalent to a non-monotone decrease of the sub-sampled objective function. Indeed, since $\{\tau_k\}$ is supposed to be a summable sequence, a stricter decrease of the sub-sampled objective function can be ensured as the number of iterations increases. We also note that the non-monotone line search mimics the step size annealing techniques frequently applied in the training of deep learning models. Indeed the presence of $\tau_k$ promotes larger learning rates during the initial iterations and reduces them as the optimization process approaches a (local) minimum point. Finally, we observe that inequality (29) can be thought of as a practical counterpart of condition (10). In (10) a summable "error" term is allowed: for this reason, we believe that inequality (28) could better reflect the nature of the theoretical requirement (10) with respect to (27). From the practical point of view, we consider the sequence

$$\tau_k = \frac{\gamma_3 \sigma_k}{\sqrt{N_k}} \varepsilon_k \tag{30}$$

using $\gamma_3 > 0$, $\sigma_k = \min\left(\sqrt{\mathrm{Var}\left(f_{\mathcal{N}_k}(x^{(k)})\right)}, \overline{\sigma}\right)$ with $\overline{\sigma} > 0$, and

$$\gamma_3 = \sqrt{2}\,\mathrm{erf}^{-1}(2\rho - 1).$$

With this selection of $\tau_k$, we take into account the lack of precision in the current approximate objective function $f_{\mathcal{N}_k}(x^{(k)})$. Indeed the quantity $\frac{\gamma_3 \sigma_k}{\sqrt{N_k}}$ is an approximate width of the confidence interval around $f_{\mathcal{N}_k}(x^{(k)})$ with $\sigma_k$ being the sample standard deviation and $\gamma_3$ the corresponding $\rho$ quantile of the Gaussian distribution $\mathcal{N}(0,1)$. Finally, in view of $N_k \leq \underline{N}$, where $\underline{N} > 0$, (see Step 4. of Algorithm 1), we remark that the following inequality holds

$$\tau_k \leq \frac{\gamma_3 \overline{\sigma}}{\sqrt{\underline{N}}} \varepsilon_k, \tag{31}$$

and hence $\{\tau_k\}_{k\in\mathbb{N}}$ is summable if also $\{\varepsilon_k\}_{k\in\mathbb{N}}$ is a summable sequence. The particular values for $\overline{\sigma}$ and $\underline{N}$ are detailed in Section 4.

**Algorithm 1** Variable metric Prox-LISA.

Given $0 < \underline{\alpha} < \overline{\alpha}$, $0 < \underline{N} < N$, $\overline{\sigma} > 0$, $\beta_1, \beta_2, \delta_1, \delta_2 \in (0,1)$, $\gamma_1, \gamma_2, \gamma_3 > 0$, and a nonnegative sequence $\{\varepsilon_k\}_{k\in\mathbb{N}}$, $\sum_{k=1}^{+\infty} \varepsilon_k < +\infty$. Select initial values $x^{(1)} \in \mathbb{R}^d$, $\alpha_1 \in (\underline{\alpha}, \overline{\alpha})$, $V^{(0)} = V_{\text{var}}^{(0)} = 0$, $\overline{V}^{(1)} = \gamma_1$, null vectors $g^{(0)}, g_{\text{var}}^{(0)} \in \mathbb{R}^d$, and $N_1 = \underline{N}$.

FOR $k = 1, 2, \ldots$

STEP 1. *Sample selection*

choose a sample $\mathcal{N}_k$ of size $N_k$ and compute its gradient $g_{\mathcal{N}_k}(x^{(k)})$ and variance $V_{\mathcal{N}_k}(x^{(k)})$.
IF $V_{\mathcal{N}_k}(x^{(k)}) \le \overline{V}^{(k)}$ OR $N_k \ge N$
THEN go to STEP 2.
ELSE set $N_k = \min\left\{N, \max\left\{\frac{N_k V_{\mathcal{N}_k}(x^{(k)})}{\overline{V}^{(k)}}, N_k + 1\right\}\right\}$ and go to STEP 1.

STEP 2. *Update running statistics and scaling matrix*
Compute the running statistics of the variance estimate and update the variance upper bound

$$V^{(k)} = \beta_1 V^{(k-1)} + (1-\beta_1) V_{\mathcal{N}_k}(x^{(k)})$$
$$V_{\text{var}}^{(k)} = \beta_2 V_{\text{var}}^{(k-1)} + (1-\beta_2)(V_{\mathcal{N}_k}(x^{(k)}) - V^{(k)})^2$$
$$\overline{V}^{(k+1)} = \min\left(\gamma_1 \varepsilon_k, \frac{V^{(k)}}{1-\beta_1^k} + \gamma_2 \sqrt{\frac{V_{\text{var}}^{(k)}}{1-\beta_2^k}}\right);$$

and the gradient estimate

$$g^{(k)} = \beta_1 g^{(k-1)} + (1-\beta_1) g_{\mathcal{N}_k}(x^{(k)})$$
$$g_{\text{var}}^{(k)} = \beta_2 g_{\text{var}}^{(k-1)} + (1-\beta_2)(g_{\mathcal{N}_k} - g^{(k)}(x^{(k)}))^2$$

to compute the diagonal scaling matrix $D_k$ according to (23).

STEP 3. *Step size selection*

Let $\bar{x}^{(k)} = \text{prox}_{\alpha_k R}^{D_k}(x^{(k)} - \alpha_k D_k^{-1} g_{\mathcal{N}_k}(x^{(k)}))$, and $\tau_k$ as in (30).
IF

$$f_{\mathcal{N}_k}(\bar{x}^{(k)}) \le f_{\mathcal{N}_k}(x^{(k)}) + g_{\mathcal{N}_k}(x^{(k)})^T(\bar{x}^{(k)} - x^{(k)}) + \frac{1}{2\alpha_k}\|\bar{x}^{(k)} - x^{(k)}\|_{D_k}^2 + \tau_k$$

THEN go to STEP 4.
ELSE set $\alpha_k \leftarrow \delta_1 \alpha_k$ and repeat STEP 3.

STEP 4. *Sample and step size prolongation*
Set $x^{(k+1)} = \bar{x}^{(k)}$, $\alpha_{k+1} = \min\left(\overline{\alpha}, \max\left(\frac{\alpha_k}{\delta_2}, \underline{\alpha}\right)\right)$, and $N_{k+1} = \max(\lfloor N_k \delta_2 \rfloor, \underline{N})$

END FOR

## 4. Numerical experiments

In this section, we evaluate the performance of the proposed proximal scaled stochastic gradient methods, denoted by Prox-LISA-VM. We compare our method to stochastic gradient descent (SG) and proximal LISA (prox-LISA) [20] on binary classification tasks using simple convex statistical models. In addition, we consider more challenging multi-class image classification problems by learning deep neural networks using Prox-LISA-VM and compare its performance to the Adam [35] and Adabelief [32] optimization schemes.

Throughout all numerical experiments, the following hyperparameters are used for Prox-LISA-VM. We set the positive and summable sequence $\varepsilon_k$ to

$$\varepsilon_k = \exp\left(-\frac{k^2}{2\sigma^2}\right)$$

for $\sigma > 0$ due to a slower initial decrease compared to $\exp(-k)$. In practice, we set $\sigma^2 = -\frac{K^2}{2\log(1/100)}$ using the maximal number of steps $K$. Likewise, we set the scaling factor of the $\varepsilon_k$-sequence to $\gamma_1 = 10^4$ and $\gamma_2 = 4$ for determining the sample size. The scaling factor for the non-monotone line search is set to

$$\gamma_3 = \sqrt{2}\text{erf}^{-1}(2\rho - 1),$$

to reflect the $\rho = 0.75$ quantile of the $\mathcal{N}_k$ sample's objective values. We use the smoothing parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ to estimate the running statistics, and $\epsilon = 10^{-16}$, which are also the default parameters in [32]. To fulfill the constraint for the line search, we choose a large enough upper bound $\overline{\sigma} = 10^6$ for the $\sigma_k$ sequence and set $\underline{\alpha} = 10^{-10}$ and $\overline{\alpha} = 10^{10}$. Finally, we set the learning rate factors of the line search $\delta_1 = \delta_2 = \frac{2}{3}$.

**Table 1**
Features of the data sets considered in the convex binary classification setting.

| Data set | $d$ | $N$ training set | $N$ test set |
|---|---|---|---|
| *MNIST* | 784 | 60000 | 10000 |
| *w8a* | 300 | 44774 | 4975 |
| *CHINA0* | 132 | 16033 | 1604 |
| *GISETTE* | 5000 | 6000 | 1000 |
| *IJCNN1* | 22 | 49990 | 91701 |
| *RCV1* | 47236 | 20242 | 10000 |

### 4.1. Convex setting

We consider the optimization problem arising in training binary classifier, with the form

$$\min_{x\in\mathbb{R}^d} P(x) = \min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} f_i(x) + \lambda\|x\|_1,$$

where $\lambda > 0$ is the regularization parameter, $\lambda = \frac{1}{N}$ where $N$ is the number of samples in the training set.
We consider six data sets and two different convex loss functions. Table 1 shows the details of the six data sets and the number of samples of both the training and the testing sets. The data sets w8a, GISETTE, IJCNN1, RCV1 can be found in the same repository,[3] whereas MNIST[4] and CHINA0[5] are available online. In particular, we adapt *MNIST* for the binary case. The two classes are the even and odd digits.

We built linear classifiers corresponding to two different convex loss functions. By denoting as $\xi_i \in \mathbb{R}^d$ and $\zeta_i \in \{1, -1\}$ the feature vector and the class label of the $i$-th example, respectively, the loss function $F(x)$ assumes one of the following forms:

- logistic regression (LR) loss:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} \log\left[1 + e^{-\zeta_i \xi_i^T x}\right];$$

- square loss (SL):

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} (1 - \zeta_i \xi_i^T x)^2;$$

In this section, we compare the behavior of:

- Prox-SG;
- Prox-LISA (as defined in [20]);
- Prox-LISA $\varepsilon_k = \exp\left(-\frac{k^2}{2\sigma^2}\right)$ (as in [20] but using $\varepsilon_k = \exp\left(-\frac{k^2}{2\sigma^2}\right)$);
- Prox-LISA-VM.

For the Prox-SG method, we consider a fixed sample size $N_k = B$ and a decreasing learning rate sequence. In particular, for all the test problems, we set $B = 50$ and $\alpha_j = \frac{100\alpha_1}{100+j}$, $j \geq 0$, where $\alpha_1$ is the initial learning rate and $j$ denotes the counter of the epochs. We select the initial learning rate as $\alpha_1 = \alpha_{opt} \cdot \overline{N}$, where $\alpha_{opt}$ is the best-tuned value for the initial learning rate found for Prox-SG with $B = 1$. We remark that the value for $\alpha_{opt}$ has been obtained through time and resource-consuming procedure of repeated trials. For all the other methods we consider $N_1 = \underline{N} = 32$ and $\alpha_1 = 1e-5$. Furthermore, to fulfill the assumptions required for the stronger convergence guarantees of Theorem 3, we use $\nu_k = \frac{10^{10}}{k^2}$. We recall that for Prox-LISA the sequence $\varepsilon_k$ has been fixed as $100 \cdot 0.999^k$ in [20]. To determine whether the benefits achieved by applying Prox-LISA-VM result solely from the selection of a new sequence $\varepsilon_k$, we also consider the original version of Prox-LISA but with the sole variation being the selection of $\varepsilon_k$.

In Figs. 1-2, we report the average graph of the optimality gap. We refer to these as average graphs since we conducted the test five times using different pseudo-random number generators, and the displayed graphs represent the averaged results. In stochastic contexts, it is good practice to analyze different realizations to get statistically robust and meaningful outcomes. The x-axis represents the epochs, where an epoch can be defined as a single pass through the training set. On the other hand, the y-axis represents the

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/.
[4] https://yann.lecun.com/exdb/mnist/.
[5] https://www.causality.inf.ethz.ch/home.php.

**Table 2**
Accuracy for all the data sets with Logistic Regression loss function.

| | MNIST | w8a | CINA0 | GISETTE | IJCNN1 | RCV1 |
|---|---|---|---|---|---|---|
| mean | 0.8988 | 0.9062 | 0.9210 | 0.9818 | 0.9206 | 0.9572 |
| std | 0.0004 | 0.0011 | 0.0024 | 0.0047 | 0.0024 | 0.0048 |
| min | 0.8961 | 0.9053 | 0.9168 | 0.9757 | 0.9113 | 0.9422 |
| max | 0.9006 | 0.9079 | 0.9239 | 0.986 | 0.9232 | 0.9668 |

**Table 3**
Optimality gap for all the data sets with Logistic Regression loss function.

| | MNIST | w8a | CINA0 | GISETTE | IJCNN1 | RCV1 |
|---|---|---|---|---|---|---|
| mean | 0.0049 | 0.0036 | 0.0024 | 0.0194 | 0.0006 | 0.0121 |
| std | 0.0013 | 0.0011 | 0.0001 | 0.0043 | 0.0004 | 0.0036 |
| min | 0.0032 | 0.0020 | 0.0014 | 0.0137 | 0.0002 | 0.0092 |
| max | 0.0077 | 0.0061 | 0.0052 | 0.0275 | 0.0024 | 0.0168 |

**Table 4**
Accuracy for all the data sets with Square Loss function.

| | MNIST | w8a | CINA0 | GISETTE | IJCNN1 | RCV1 |
|---|---|---|---|---|---|---|
| mean | 0.8936 | 0.8919 | 0.9196 | 0.9789 | 0.9108 | 0.9468 |
| std | 0.0009 | 0.0085 | 0.0017 | 0.0026 | 0.0003 | 0.0038 |
| min | 0.8906 | 0.8899 | 0.9148 | 0.9732 | 0.9101 | 0.9422 |
| max | 0.8964 | 0.8947 | 0.9221 | 0.984 | 0.9119 | 0.9631 |

**Table 5**
Optimality gap for all the data sets with Square Loss function.

| | MNIST | w8a | CINA0 | GISETTE | IJCNN1 | RCV1 |
|---|---|---|---|---|---|---|
| mean | 0.0027 | 0.0009 | 0.0024 | 0.0577 | 0.0003 | 0.0137 |
| std | 0.0003 | 0.0002 | 0.0001 | 0.0028 | 0.0001 | 0.0021 |
| min | 0.002 | 0.0005 | 0.0014 | 0.0211 | 0.0001 | 0.0103 |
| max | 0.0034 | 0.0024 | 0.0052 | 0.0639 | 0.0007 | 0.0148 |

values of the optimality gap. To obtain a good estimate of $P^*$ we performed Prox-SG for 3000 epochs. As Figs. 1-2 show, the proposed method outperforms the others in six out of twelve cases, demonstrating the significant acceleration of the learning process in the analyzed binary classification scenarios. In the remaining tests, where the improvement is less evident, it is worth noting that the proposed method exhibits comparable performance to existing methods from the literature.

#### *4.1.1. Ablation of hyperparameters*

To evaluate the impact of the different hyperparameters of Prox-LISA-VM, we conducted an ablation study. Specifically, we examined the following parameters: $\underline{N} \in \{10, 32, 64\}$, $\delta_1 \in \{1/2, 2/3\}$, $\delta_2 \in \{1/2, 2/3\}$, $\gamma_2 \in \{3, 4, 5\}$, and $\gamma_3 \in \{0.60, 0.75, 0.9\}$, resulting in a total of 108 hyperparameter configurations. Tables 2–5 list the mean and the standard deviation of the final accuracy (on the test set) and the final value of the optimality gap (on the training set) over the 108 configurations. Moreover, the minimum and the maximum values obtained for both the accuracy and the optimality gap are also provided. The results of this ablation study provide strong evidence that the algorithm maintains stability across a range of settings for these hyperparameters. This robustness is further confirmed by the fact that in all the proposed numerical experiments, we kept the same configuration for all the hyperparameters.

We excluded the hyperparameters $\beta_1$, $\beta_2$, and $\epsilon$ from the ablation study. This decision was made because we opted for standard values for these hyperparameters, which align with the recommendations for Adabelief. The remaining parameters $(\underline{\alpha}, \overline{\alpha}, \overline{\sigma})$ were not subjected to ablation. These parameters play a critical role in the theoretical results of our approach, although their values do not significantly impact the algorithm's performance. Particularly we select $\underline{\alpha} = 10^{-10}, \overline{\alpha} = 10^{10}$ and $\overline{\sigma} = 10^6$ to force that they are never attained by $\alpha_k$ and $\sigma_k$ throughout the iterative process.

### *4.2. Non-convex setting*

Let $\mathcal{D} = \{\xi_i, \zeta_i\}_{i=1}^N$ denote the training data set consisting of inputs $\xi_i \in \mathbb{R}^n$ and corresponding one-hot encoded class labels $\zeta_i \in \Delta^C$, where $\Delta^C = \{\zeta \in \mathbb{R}^C : \zeta_c \geq 0, \sum_{c=1}^C \zeta_c = 1\}$ is the unit simplex. Further, let $G : \mathbb{R}^n \times \mathbb{R}^d \to \Delta^C$ be a (deep) neural network that predicts a distribution over class labels $\hat{\zeta}_i \in \Delta^C$ for a given input $\xi_i \in \mathbb{R}^n$ and is parameterized by $x \in \mathbb{R}^d$. Then, every component $f_i : \mathbb{R}^d \to \mathbb{R}$ of the corresponding empirical risk reads as

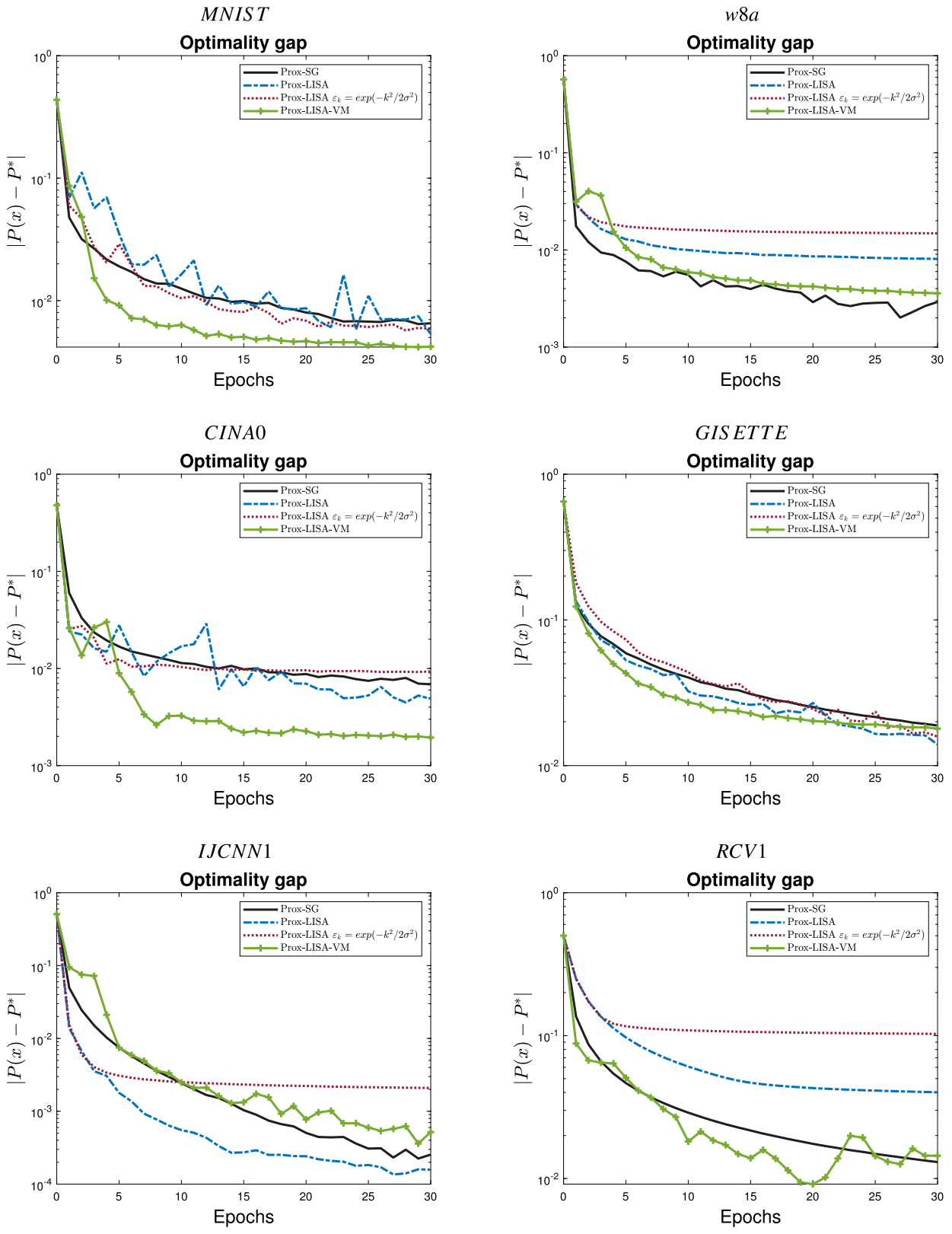


**Fig. 1.** Binary classification with the Logistic Regression function on 5 different trials.

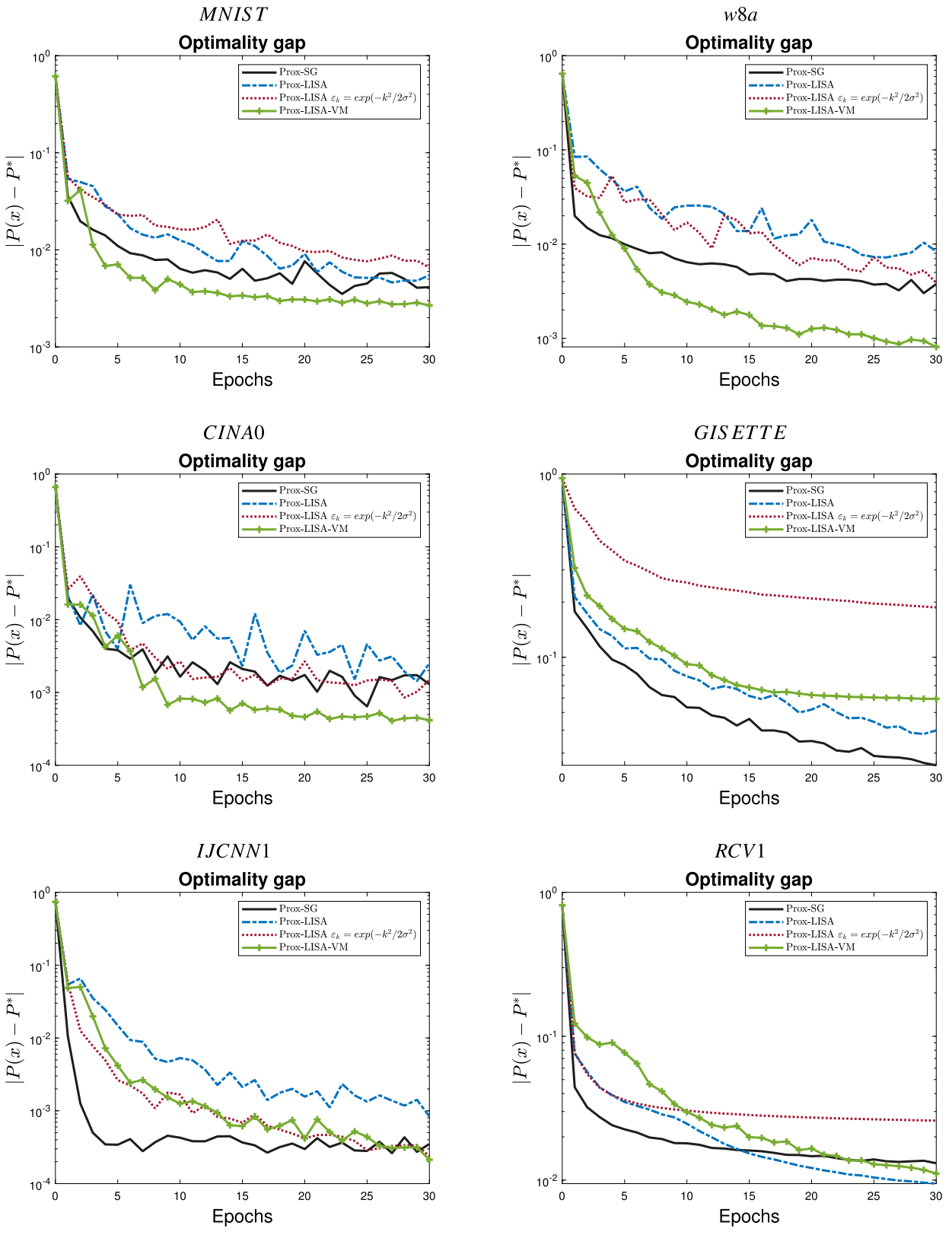


**Fig. 2.** Binary classification with the Square Loss function on 5 different trials.

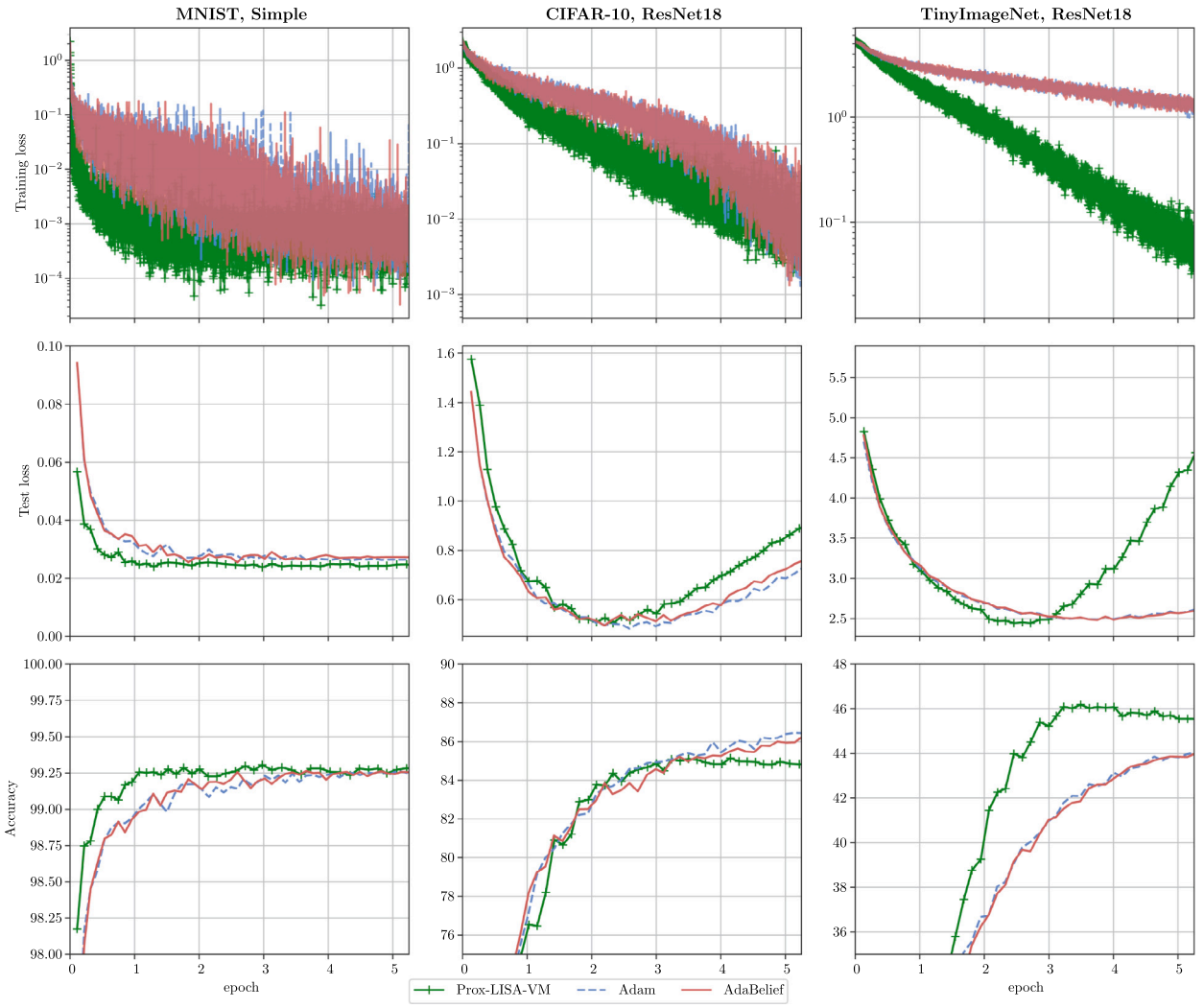


**Fig. 3.** Training and test loss as well as test accuracy for the considered non-convex multi-class classification problems.

$$f_i(x) = \ell(G(\xi_i, x), \zeta_i),$$

where $\ell : \Delta^C \times \Delta^C \to \mathbb{R}_+$ is the cross-entropy loss, i.e.

$$\ell(\hat{\zeta}, \zeta) = -\sum_{c=1}^{C} \zeta_i \log(\hat{\zeta}_i).$$

In this setting, we evaluate our optimization method for the following non-convex optimization problems:

- **MNIST Simple:** In this case we use a simple feed-forward network to classify grayscale images of size $28 \times 28$ ($n = 784$) of the MNIST data set into the depicted digit ($C = 10$). In detail, the neural network $G$ consists of two layers that successively perform a convolution, ReLU activation function, and $2 \times 2$ max-pooling using 64 and 32 feature channels, respectively. A final fully connected linear layer along with a softmax activation function maps the intermediate features to predictions $\hat{\zeta}_i$.
- **CIFAR-10 ResNet18:** For this problem, the CIFAR-10 data set [36] is used, which consists of 60 000 RGB-images of size $32 \times 32$ ($n = 3072$) belonging to $C = 10$ different classes. The training set and the test set contain 50 000 and 10 000 images, respectively. We use the ResNet18 [37] model as neural network $G$, which contains $d = 11\ 169\ 162$ trainable parameters. To avoid any side effects due to batch normalization, we removed those layers from the model.
- **TinyImageNet ResNet18:** For the last test, the same neural network as in the previous setting (ResNet18) is used. However, we consider a more challenging image classification problem using the TinyImageNet data set [38], which is a reduced version of the well-known ImageNet data set [39]. In detail, the TinyImageNet data set consists of 100 000 train and 10 000 test RGB images

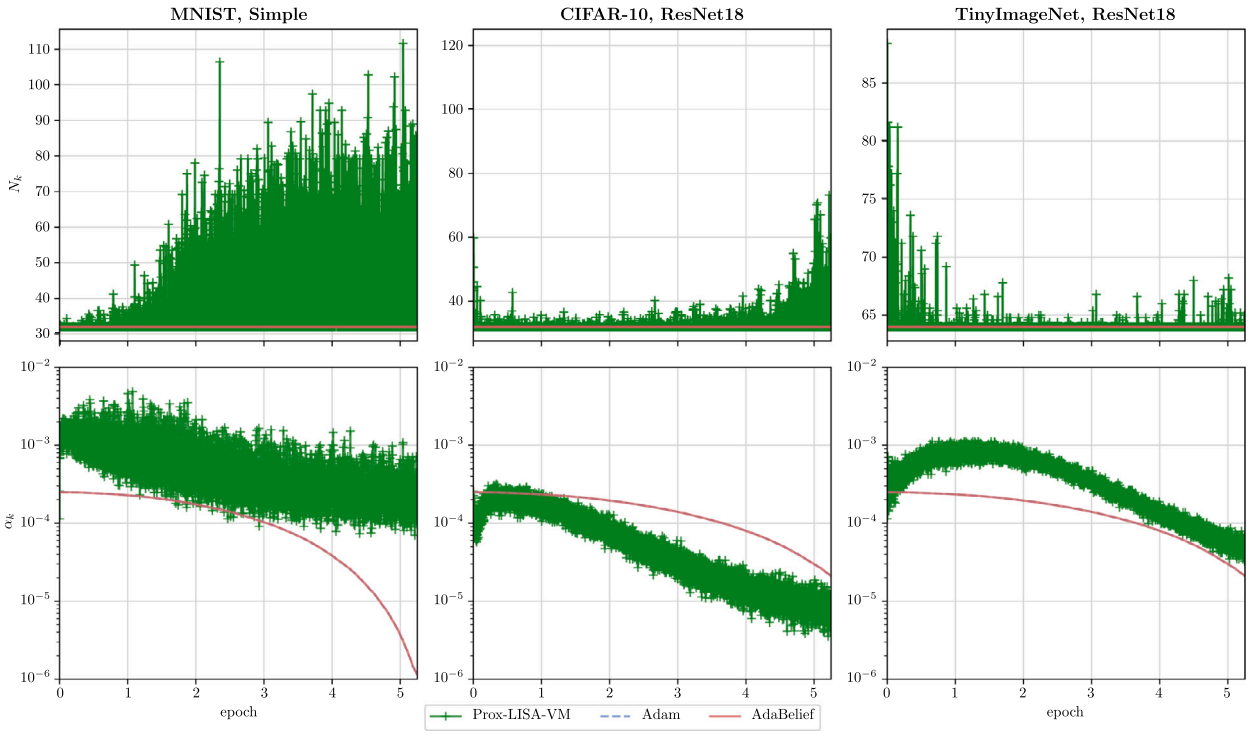


**Fig. 4.** Sample size $N_k$ and step size $\alpha_k$ over the epochs for the three considered non-convex deep learning problems.

of size $64 \times 64$ ($n = 12\,288$), which depict instances of $C = 200$ categories. In this case, the number of trainable parameters of the neural network is 11 266 632.

To minimize the effect of overfitting, we utilize weight decay regularization in all cases, that is $R(x) = \frac{\lambda}{2}\|x\|_2^2$. For the three different setting $\lambda$ is set to $1 \times 10^{-4}$, $5 \times 10^{-4}$, and $1 \times 10^{-4}$, respectively. As the lower bound for the sample set size, we used $\underline{N} = 32$ for the MNIST and CIFAR-10 data set and $\underline{N} = 64$ for the TinyImageNet data set to account for the increased initial variance.

In all tests, we compared our algorithm with the Adam [35] and Adabelief [32] optimization schemes, which already proved its competitiveness to the standard stochastic gradient descent schemes. For both algorithms, we used a fixed sample size $N_k = \underline{N}$ and set the initial learning rate to $2.5 \times 10^{-4}$ and reduced it to $1 \times 10^{-6}$ using cosine annealing. The remaining hyper-parameters are set to the default values. This setup led to the fastest convergence for all three considered problems.

Fig. 3 depicts for all three considered problems from top to bottom the sampled training loss $f_{\mathcal{N}_k}(x)$, the loss on the test set, and the accuracy on the test set as a function of epochs. Given the stochasticity of the sampling, all the metrics are averaged over 5 different runs, as in the convex setting. Concerning the training loss, Prox-LISA-VM shows a faster decrease rate compared to Adam and Adabelief, especially for the TinyImageNet data set and in the initial phase. However, the overall accuracy is comparable for both methods across the different problems. We highlight that for Prox-LISA-VM no selection of the learning rate is required; only a reasonable selection of the lower bound of the sample size $\underline{N}$ is necessary to allow a reliable estimation of the sample's statistics.

The average sample size $N_k$ along with the average step size $\alpha_k$ across 5 different runs is depicted in Fig. 4. In all tests, we observe that the sample size is increasing and decreasing. For the MNIST data set the sample size has a tendency to increase toward the end, while for the CIFAR-10 and TinyImageNet data set the sample size is only slightly larger initially and at the end. The plots of the step sizes in the second row show similar behavior in all three tests; there is an initial increase followed by a decrease of about an order of magnitude toward the end of the iterative process. This decrease originates from the reduced non-monotonicity of the line search due to a decreasing $\varepsilon_k$ sequence.

Finally, we would like to compare the computational complexity of the different algorithms. While Prox-SG, Adam, and Adabelief only require the estimation of a sample's objective and gradient, Prox-LISA-VM requires the estimation of the objective value and gradient for every sample to determine the sample size $N_k$. However, the computational complexity of both tasks is the same for a given sample size. Moreover, the experimental results demonstrate that Prox-LISA-VM performs comparably to the state-of-the-art if epochs, which do account for the varying sample size, are considered. Further, the average number of backtracking steps in the line search procedure was at most 2 for all considered problems. Thus, the computational complexity of Prox-LISA-VM is similar to the related methods.

## 5. Conclusions

In this paper, we presented a variable metric approach for preconditioning stochastic gradient directions in conjunction with an automatic sample size selection to control the variance of stochastic gradient directions for regularized empirical risk minimization problems. We developed conditions for the sequence of variable metrics to ensure convergence in convex and non-convex settings. Various numerical experiments demonstrated that the proposed method performs equally well or even outperforms state-of-the-art methods on challenging binary and multi-class classification problems. Moreover, our proposed method only requires a rough selection of the sample size lower bound and no further hyperparameters need to be adapted due to the utilization of a non-monotone line search criterion.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was partially supported by “Gruppo Nazionale per il Calcolo Scientifico (GNCS- INdAM)” (Progetto 2023 “Modelli e metodi avanzati in Computer Vision”). The publication was created with the co-financing of the European Union-FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062/2021.
F. Porta is supported by the project PNRR - Missione 4 “Istruzione e Ricerca” - Componente C2 “Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN)” “Numerical Optimization with Adaptive Accuracy and Applications to Machine Learning”, project code: 2022N3ZNAX (CUP E53D23007700006) funded by the European Commission under the NextGeneration EU programme.
F. Porta and A. Sebastiani are supported by the project “PNRR - Missione 4 “Istruzione e Ricerca” - Componente C2 Investimento 1.1 “Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN)”, “Advanced optimization METhods for automated central veIn Sign detection in multiple sclerosis from magneTic resonAnce imaging (AMETISTA)”, project code: P2022J9SNP, MUR D.D. financing decree n. 1379 of 1st September 2023 (CUP E53D23017980001) funded by the European Commission under the NextGeneration EU programme.

## References

[1] C. Li, X. Li, M. Chen, X. Sun, Deep learning and image recognition, in: 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT), IEEE, 2023, pp. 557–562.
[2] G. Varshney, M. Misra, P.K. Atrey, A survey and classification of web phishing detection schemes, Secur. Commun. Netw. 9 (18) (2016) 6266–6284.
[3] M. Bakator, D. Radosav, Deep learning and medical diagnosis: a review of literature, Multimod. Technol. Interact. 2 (3) (2018) 47.
[4] L.B. Reller, M.P. Weinstein, C.A. Petti, Detection and identification of microorganisms by gene amplification and sequencing, Clin. Infect. Dis. 44 (8) (2007) 1108–1114.
[5] P. Cascarano, M.C. Comes, A. Mencattini, M.C. Parrini, E.L. Piccolomini, E. Martinelli, Recursive deep prior video: a super resolution algorithm for time-lapse microscopy of organ-on-chip experiments, Med. Image Anal. 72 (2021) 102124.
[6] P. Cascarano, M.C. Comes, A. Sebastiani, A. Mencattini, E. Loli Piccolomini, E. Martinelli, Deepcel0 for 2d single-molecule localization in fluorescence microscopy, Bioinformatics 38 (5) (2022) 1411–1419.
[7] P.L. Combettes, R. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model. Simul. 4 (2005) 1168–1200.
[8] P.L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: H.H. Bauschke, R.S. Burachik, V. Combettes, P.L. Elser, D.R. Luke, H. Wolkowicz (Eds.), Fixed-Point Algorithms for Inverse Problems in Science and Engineering, in: Springer Optim. Appl., vol. 49, Springer, New York, 2011, pp. 185–212.
[9] S.L. Smith, P.-J. Kindermans, C. Ying, Q.V. Le, Don’t decay the learning rate, increase the batch size, preprint, arXiv:1711.00489.
[10] A. Devarakonda, M. Naumov, M. Garland, Adabatch: adaptive batch sizes for training deep neural networks, preprint, arXiv:1712.02029.
[11] R.M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, P. Richtárik, Sgd: general analysis and improved rates, in: International Conference on Machine Learning, PMLR, 2019, pp. 5200–5209.
[12] S.L. Smith, Q.V. Le, A bayesian perspective on generalization and stochastic gradient descent, preprint, arXiv:1710.06451.
[13] L. Xiao, T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM J. Optim. 24 (4) (2014) 2057–2075.
[14] N.H. Phamy, L.M. Nguyen, D.T. Phan, Q. Tran-Dinh, Proxsarah: an efficient algorithmic framework for stochastic composite nonconvex optimization, J. Mach. Learn. Res. 21 (2020) 1–48.
[15] R. Bollapragada, R. Byrd, J. Nocedal, Adaptive sampling strategies for stochastic optimization, SIAM J. Optim. 28 (4) (2018) 3312–3343.
[16] L. Bottou, F.E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. 60 (2) (2018) 223–311.
[17] R.H. Byrd, G.M. Chin, J. Nocedal, Y. Wu, Sample size selection in optimization methods for machine learning, Math. Program. 1 (134) (2012) 127–155.
[18] F.H. Hashemi, S. Ghosh, R. Pasupathy, On adaptive sampling rules for stochastic recursions, in: Simulation Conference (WSC), 2014 Winter, 2014, pp. 3959–3970.
[19] A. Cutkosky, F. Orabona, Momentum-based variance reduction in non-convex sgd, Adv. Neural Inf. Process. Syst. 32.
[20] G. Franchini, F. Porta, I. Trombini, V. Ruggiero, A line search based proximal stochastic gradient algorithm with dynamical variance reduction, J. Sci. Comput. 94 (2023) 23.
[21] S. Bonettini, I. Loris, F. Porta, M. Prato, Variable metric inexact line-search based methods for nonsmooth optimization, SIAM J. Optim. 26 (2016) 891–921.
[22] E. Chouzenoux, J.-C. Pesquet, A. Repetti, Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function, J. Optim. Theory Appl. 162 (2014) 107–132.
[23] P.L. Combettes, B. Vũ, Variable metric forward-backward splitting with applications to monotone inclusions in duality, Optimization 63 (2014) 1289–1318.
[24] P. Frankel, G. Garrigos, J. Peypouquet, Splitting methods with variable metric for Kurdyka-Łojasiewicz functions and general convergence rates, J. Optim. Theory Appl. 165 (2015) 874–900.
[25] G. Franchini, F. Porta, F. Ruggiero, I. Trombini, L. Zanni, Learning rate selection in stochastic gradient methods based on line search strategies, Appl. Math. Sci. Eng. 31 (1) (2023) 2164000.

[26] N. Krejić, N. Krklec Jerinkić, Nonmonotone line search methods with variable sample size, Numer. Algorithms 68 (4) (2015) 711–739.
[27] B.T. Polyak, Introduction to Optimization, Optimization Software, New York, 1987.
[28] P. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, H. Wolkowicz (Eds.), Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer Optimization and Its Applications, Springer, New York, NY, 2011, pp. 185–212.
[29] P.L. Combettes, B. Vũ, Variable metric quasi-Féjer monotonicity, Nonlinear Anal. 78 (2013) 17–31.
[30] G. Franchini, F. Porta, V. Ruggiero, I. Trombini, Correction to: a line search based proximal stochastic gradient algorithm with dynamical variance reduction, J. Sci. Comput. 94 (1) (2023) 23.
[31] J.E. Freund, Mathematical Statistics, Prentice-Hall, Englewood Cliffs, NJ, USA, 1962.
[32] J. Zhuang, T. Tang, Y. Ding, S.C. Tatikonda, N. Dvornek, X. Papademetris, J. Duncan, Adabelief optimizer: Adapting stepsizes by the belief in observed gradients, Adv. Neural Inf. Process. Syst. 33.
[33] S. Bonettini, F. Porta, V. Ruggiero, A variable metric forward-backward method with extrapolation, SIAM J. Sci. Comput. 38 (2016) A2558–A2584.
[34] M. Schmidt, N. Le Roux, F. Bach, Minimizing finite sums with the stochastic average gradient, Math. Program. 162 (1) (2017) 83–112.
[35] D. Kingma, J. Ba Adam, A method for stochastic optimization, in: International Conference on Learning Representations (ICLR), San Diega, CA, USA, 2015.
[36] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (canadian institute for advanced research), http://www.cs.toronto.edu/~kriz/cifar.html.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[38] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, CS 231N 7 (7) (2015) 3.
[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.