Spectral Stochastic Gradient Method with Additional Sampling for Finite and Infinite Sums

Nataša Krklec Jerinkić
Faculty of Sciences, Department of
Mathematics and Informatics
University of Novi Sad
Novi Sad, Serbia
natasa.krklec@dmi.uns.ac.rs
Valeria Ruggiero
Department of Mathematics and
Computer Science
University of Ferrara
Ferrara, Italy
valeria.ruggiero@unife.it
Ilaria Trombini
Department of Mathematical, Physical
and Computer Sciences
University of Parma
Parma, Italy
Department of Mathematics and
Computer Science
University of Ferrara
Ferrara, Italy
ilaria.trombini@unife.it

October 11, 2024

Abstract

In this paper, we propose a new stochastic gradient method for numerical minimization of finite sums. We also propose a modified version of this method applicable on more general problems referred to as infinite sum problems, where the objective function is in the form of mathematical expectation. The method is based on a strategy to exploit the effectiveness of the well-known Barzilai-Borwein (BB) rules or variants of these (BB-like) rules for updating the step length in the standard gradient method. The proposed method adapts the aforementioned strategy into the stochastic framework by exploiting the same Sample Average Aproximations (SAA) estimator of the objective function for several iterations. Furthermore, the sample size is controlled by an additional sampling which also plays a role in accepting the proposed iterate point. Moreover, the number of

"inner" iterations with the same sample is also controlled by an adaptive rule which prevents the method from getting stuck with the same estimator for too long. Convergence results are discussed for the finite and infinite sum version, for general and strongly convex objective functions. For the strongly convex case, we provide convergence rate and worst-case complexity analysis. Numerical experiments on well-known datasets for binary classifications show very promising performance of the method, without the need to provide special values for hyperparameters on which the method depends.

Keywords: Stochastic gradient method, Barzilai-Borwein rules, additional sub-sampling, finite sum minimization, infinite sum minimization

1 Introduction

In this paper we consider the following unconstrained optimization problem

$$\min_{x\in\mathbb{R}^d} f(x) := \mathbb{E}[F(x,\xi)], \tag{1}$$

where $\xi \in \Omega$ is a multi-valued random variable, $F(x,\xi)$ is a cost function and the mathematical expectation $\mathbb{E}$ is defined with respect to ξ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. As $f(x)$ is rarely available analytically, one of the common approaches is to approximate the problem with the finite sum function

$$\min_{x\in\mathbb{R}^d} f_{\mathcal{N}}(x) = \frac{1}{N}\sum_{i=1}^{N} F(x,\xi_i). \tag{2}$$

We assume that $F(x,\xi_i) \equiv F_i(x)$ is a differentiable function with L_i-Lipschitz-continuous gradient. Here $f_{\mathcal{N}}(x)$ is a sample average approximation of $f(x)$, based on a fixed sample $\mathcal{N} = \{\xi_1, \ldots, \xi_{\mathcal{N}}\}$ of size N, generated at the beginning of the optimization process. In machine learning applications, $\mathcal{N}$ represents the training set. The aim of this paper is to develop a stochastic first order method, where we use a non-monotone line-search and the well-known Barzilai-Borwein (BB) rules [14] or variants of these (BB-like rules) for updating the step length along the negative stochastic gradient, which is a descent direction in expectation.

We recall that, in the full gradient iteration $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$, the standard BB rules are well-performing updating rules for the choice of α_k. They are defined as

$$\alpha_k^{BB1} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}, \qquad \alpha_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}},$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

Very effective improvements with respect to the standard BB rules have been obtained using the Adaptive Barzilai-Borwein (ABB) strategy [9] and its modification ABB_{min} [6]. In this last version the step length is defined as

$$\alpha_k^{ABB_{min}} = \begin{cases} \min\{\alpha_j^{BB2} |\ j = \max(1, k - M_\alpha), \ldots, k\} & \text{if } \frac{\alpha_k^{BB2}}{\alpha_k^{BB1}} < \tau, \\ \alpha_k^{BB1} & \text{otherwise,} \end{cases}$$

where $M_\alpha > 0$ is a prefixed integer constant and $\tau \in (0.5, 1)$. For non-quadratic minimization problems, the BB or BB-like step length of a standard gradient

method has to be projected on a prefixed, positive, arbitrary large interval $[\gamma_{min}, \gamma_{max}]$ and possibly adjusted by a line-search procedure. In many applications (see for example [17, 18, 19]), the standard gradient method (and its variants) equipped with the BB or BB-like selection rules and a monotone or non-monotone line-search strategy showed good performance.
Starting from these observations and following the approaches developed in Algorithms LSOS [10] and SLiSeS [11], in this paper we propose a stochastic gradient method where the approximated gradient is computed on a randomly chosen subset $\mathcal{N}_k$ of the available dataset, i.e., a *mini-batch* of this dataset, with a non decreasing cardinality $|\mathcal{N}_k| = N_k$. In the case of the finite sum minimization (2), we have $\mathcal{N}_k \subseteq \mathcal{N}$ and $N_k \leq N$, whereas in the infinite case $\mathcal{N}_k$ is a random sample associated with the current estimator of the objective function in (1).
The basic idea of the proposed method is to exploit the effectiveness of the standard gradient method equipped with a BB-like rule until the mini-batch $\mathcal{N}_k$ is changed. When an appropriate number of iterations (*cycle*) involving the same mini-batch were performed or a suitable Stochastic Descent (SD) condition is not met for the additional random estimator $f_{\mathcal{D}_k} = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} F(\cdot, \xi_i)$ of the objective function in (2) or (1), another mini-batch $\mathcal{N}_k$ is randomly drawn from the available dataset, by adapting the non decreasing cardinality N_k to the current scenario of the algorithm. In particular, when the cycle is successfully finished, the size of the new mini-batch is unchanged; when the SD condition is not satisfied and the cycle is unsuccessfully stopped, the size of the new mini-batch is increased. The method is described and analyzed first for the minimization of the finite sum case (2), giving rise to the scheme named **LSNM-BB**; then, it is generalized to the more general case stated in (1), leading to the version named **LSNM-BB-G**.
Recently, two methods have been proposed, LSOS and SLiSeS, for the finite sum case (2), that are similar to **LSNM-BB**. We observe that, although it is inspired by them, the **LSNM-BB** method differs in many aspects. Indeed, in **LSNM-BB** scheme, the mini-batch $\mathcal{N}_k$ does not change for a certain number of iterations, i.e., until it is possible to exploit the BB-like rules in a way that is advantageous for the original problem, giving rise to a cycle of iterations related to the same mini-batch. The effectiveness of the BB-like rules is evaluated by using the additional SD condition based on the evaluation of an additional estimator $f_{\mathcal{D}_k}$ of the objective function, where $\mathcal{D}_k$ is randomly chosen from the available dataset. In SLiSeS, the line-search procedure is very different and no additional estimator is used. Furthermore, the mini-batch $\mathcal{N}_k$ is changed after a prefixed number of iterations. On the other hand, the LSOS method is a stochastic version of the BFGS iteration; consequently, it is a second order method which involves the control of an SD condition at each iteration based on an additional estimator, as **LSNM-BB**, but when this condition is not satisfied for a prefixed maximum number of times, the method switches to predefined square summable step lengths. Unlike LSOS, **LSNM-BB** doesn't prefix a maximum number of times the SD condition is not satisfied; indeed the failure of the SD condition causes an increase of the mini-batch size. In general, in the finite sums case, $N_k < N$ holds. In some cases, it could happen that the SD condition is not met for a large number of times and, consequently, the increment rule of the mini-batch size determines a value of N_k equal to N, i.e., the mini-batch is the

whole dataset $\mathcal{N}$ and the stochastic iteration reduces to that of the standard gradient. This does not happen in practice, unless a significant increase in the cardinality of the mini-batch is forced and/or very long processing times are allowed.
Furthermore, the method proposed for the case of a finite sum of terms can be generalized to **LSNM-BB-G** for the resolution of problem (1). One of the possible applications can be to address online learning problems. In this version, the concept of mini-batch is replaced by that of estimator of f and, at any iteration, two estimators $f_{\mathcal{N}_k}$ and $f_{\mathcal{D}_k}$ are used, the first to adjust the step length, the second to verify the SD condition.
Finally, we should mention that the method proposed in [5] also uses additional sampling as a control for accepting the step and increasing the sample size if necessary. However, it differs from our approach significantly since it belongs to the Trust-Region framework, uses Hessian approximations and considers only finite sum case.
The paper is organized as follows. In Section 2, we describe the **LSNM-BB** method, giving the details of the related algorithm and stating its well-definiteness. In Section 3 we report the convergence results of the proposed method for the finite sum minimization (2), whereas in Section 4 we describe the **LSNM-BB-G** method for the problem (1) and we state the convergence results. In Section 5 we evaluate the proposed method by a set of numerical experiments. Some conclusions are drawn in the final section which also contains some future work directions.

Notation.
In the following, $\mathbb{R}_+$ is the set of non negative real numbers; $\mathbb{R}_{++}$ is the set of positive real numbers. $\|\cdot\|$ denotes the standard ℓ_2 norm. $\mathbb{E}(\cdot)$ and $\mathbb{E}(\cdot|\ \mathcal{F})$ denote mathematical expectation and conditional expectation with respect to σ-algebra $\mathcal{F}$, respectively. We use "a.s." to abbreviate "almost sure/surely" and "i.i.d." to abbreviate "independent and identically distributed", while "SAA" stands for "sample average approximation". We denote by $|\mathcal{N}|$ the cardinality of set $\mathcal{N}$. Finally, $\mathcal{B}(x,\rho)$ denotes the ball of center x and radius ρ.

2 The LSNM-BB method for the finite sum case

The **LSNM-BB** method is a first-order scheme, consisting of cycles of iterations. Each j-th cycle can have a maximum number of iterations, denoted by $m(N_j)$, characterized by the use of the same mini-batch $\mathcal{N}_j$ of the dataset $\mathcal{N}$, where $\mathcal{N}_j \equiv \mathcal{N}_k$ for any iteration k in the cycle. The search directions are obtained by combining a suitable version of BB rules with the approximations of the gradient based on $\mathcal{N}_k$. In other words, within any cycle, the method boils down to the gradient descent iteration for $f_{\mathcal{N}_k}(x)$ combined with a non-monotone line-search to adjust the step length, fixed by a BB-like rule. The cycle can be stopped prematurely if it does not produce iterations deemed acceptable. Indeed, after the computation of the update $\overline{x}_k$ of a standard gradient iteration for $f_{\mathcal{N}_k}$, the following SD condition is checked:

$$f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k, \tag{3}$$

where $\mathcal{D}_k$ is randomly chosen from $\mathcal{N}$, c_{min}, C_{max} are positive real scalars and $\{\zeta_k\}$ is a summable sequence of non-negative real numbers, so that the strict decrease of $f_{\mathcal{D}_k}$ is relaxed. If the above condition is not met, the current cycle is stopped; the vector $\overline{x}_k$ is rejected and $x_{k+1} = x_k$; a new cycle is started using a new mini-batch $\mathcal{N}_{k+1}$ of larger size ($N_k < N_{k+1} \leq N$). The step length is set as the tentative value $\gamma_{k+1} = \frac{1}{\|\nabla f_{\mathcal{N}_{k+1}}(x_{k+1})\|}$, suitably projected on $[\gamma_{min}, \gamma_{max}]$. The choice to adopt at the first step of a new cycle the scaled stochastic gradient is convenient in the practice, but it is not relevant for the theoretical analysis; other setting for γ_{k+1} can be performed, as $\gamma_{k+1} = 1$ for instance.
Conversely, when the SD condition (3) is met at $\overline{x}_k$, then $x_{k+1} = \overline{x}_k$, the initial step length is updated by a new BB-like rule and a new iteration of the cycle is performed.
If the j-th cycle ends by completing the foreseen $m(N_j)$ iterations, a new mini-batch is randomly extracted from $\mathcal{N}$, leaving its cardinality equal to that of the previous mini-batch, that is $N_{j+1} = N_j$; then, a new cycle is started, aimed to perform a set of iterations which lead to decrease of a new approximation $f_{\mathcal{N}_{j+1}}$ of $f_{\mathcal{N}}$.

The details of the proposed **LSNM-BB** method are described in Algorithm 1. In particular, as already specified, within any cycle, the search directions d_k is computed as $d_k = -\gamma_k g_k$ where g_k is the gradient of the current estimate of the objective function at x_k and γ_k is initialized with a BB-like rule, suitably projected on $[\gamma_{min}, \gamma_{max}]$. An adjustment of the step length, given by $t_k = \beta^{\ell}$, $\beta \in (0, 1)$, $\ell \in \mathbb{N}$, is determined by a line-search technique, aimed at ensuring that the non-monotone Armijo condition (4) is satisfied. A key assumption that makes the algorithm well-defined is the following.

Assumption 1. *For any $i \in \{1, ..., N\}$, the function F_i is bounded from below and continuously-differentiable with L_i-Lipschitz continuous gradient.*

Consequently, $L_{\mathcal{N}_k} = \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} L_i$ is the Lipschitz parameter of $\nabla f_{\mathcal{N}_k}$ and $L_{\mathcal{N}_k} \in [L_{\mathcal{N}}, L_{max}]$, with $L_{\mathcal{N}}$ equal to the Lipschitz parameter of $\nabla f_{\mathcal{N}}$ and $L_{max} = \max_i L_i$; obviously, $L_{\mathcal{N}_k} \leq L_{max}$.

Remark 1. *In view of the above assumption, it is well known that the standard monotone line-search technique is well-defined, that is there exists a value $\overline{t} = \beta^{\overline{\ell}}$ with $\overline{t} \geq \min(1, \frac{2\beta(1-\eta)}{L_{\mathcal{N}_k}})$ such that, for $\gamma_k \in [\gamma_{min}, \gamma_{max}]$, the following condition is met*

$$f_{\mathcal{N}_k}(x_k - \overline{t}\gamma_k g_k) \leq f_{\mathcal{N}_k}(x_k) - \eta \overline{t} \gamma_k \|g_k\|^2. \tag{5}$$

Thus, the value $\overline{t}$ is bounded from below by $t_{min} = \min(1, \frac{2\beta(1-\eta)}{L_{max}})$. The term $\zeta_k \geq 0$ in the condition (4) can be positive, allowing for nondescent directions and relaxing the condition (5) ; for t_k small enough, the condition (4) is satisfied and hence the finite termination of the backtracking loop is assured.

The further following assumption is required.

Assumption 2. *The non-negative real sequence $\{\zeta_k\}$ in (3) and in (4) is such that $\sum_{k=0}^{\infty} \zeta_k \leq \overline{\zeta}$.*

Remark 2. *When d_k satisfies the SD condition (3), we notice that the reduction of $f_{\mathcal{D}_k}$, although relaxed by the presence of $C_{max}\zeta_k \geq 0$, can be considered as*

Algorithm 1 LSNM-BB

Given $x_0 \in \mathbb{R}^d$, $\eta, \beta \in (0,1)$, $\{\zeta_k\} \subset \mathbb{R}_+$ subject to $\sum_{k=0}^{\infty} \zeta_k \leq \overline{\zeta} < \infty$, c_{min}, $C_{max} \in \mathbb{R}_{++}$, $0 < \gamma_{min} < \gamma_{max}$, $\gamma_0 \in [\gamma_{min}, \gamma_{max}]$, $N_0 > 0$, $\theta > 1$.
Set $k \leftarrow 0$, $j \leftarrow 0$.
Choose $\mathcal{N}_0 \subseteq \mathcal{N}$ randomly with size N_0
Compute $g_0 \leftarrow \nabla f_{\mathcal{N}_0}(x_0)$
while the stopping criterion is not satisfied **do**
    Compute $m(N_j)$
    $i = 1$
    repeat
        Compute $d_k \leftarrow -\gamma_k g_k$
        Find the smallest integer $\ell \geq 0$ such that $t_k = \beta^\ell$ satisfies

$$f_{\mathcal{N}_k}(x_k + t_k d_k) \leq f_{\mathcal{N}_k}(x_k) + \eta t_k g_k^T d_k + \zeta_k \tag{4}$$

        $\overline{x}_k \leftarrow x_k + t_k d_k$
        if $N_k < N$ **then**
            Choose $\mathcal{D}_k$ randomly and uniformly from $\mathcal{N}$ with replacement
            if $f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k$ **then**
                $x_{k+1} \leftarrow \overline{x}_k$
                $\mathcal{N}_{k+1} \leftarrow \mathcal{N}_k$
                $N_{k+1} \leftarrow N_k$
                Compute $g_{k+1} \leftarrow \nabla f_{\mathcal{N}_{k+1}}(x_{k+1})$
                Compute γ_{k+1} by a BB-like rule with threshold in $[\gamma_{min}, \gamma_{max}]$
            else
                $x_{k+1} \leftarrow x_k$
                Choose $N_{k+1} \in (N_k, N]$
                $k \leftarrow k + 1$
                Exit and go to step 34
            end if
        else $(\mathcal{N}_k = \mathcal{N})$
            $x_{k+1} \leftarrow \overline{x}_k$
            Compute $g_{k+1} \leftarrow \nabla f_{\mathcal{N}}(x_{k+1})$
            Compute γ_{k+1} by a BB-like rule with threshold in $[\gamma_{min}, \gamma_{max}]$
        end if
        $k \leftarrow k + 1$
        $i = i + 1$
    until $i > m(N_j)$ OR the stopping criterion is satisfied
    if $N_k < N$ **then**
        Randomly choose $\mathcal{N}_k \subseteq \mathcal{N}$ with size N_k
        Compute $g_k \leftarrow \nabla f_{N_k}(x_k)$
        Compute γ_k as $\frac{1}{\|g_k\|}$ with threshold in $[\gamma_{min}, \gamma_{max}]$
    end if
    $j \leftarrow k$
end while

an indication that the decrease of $f_{\mathcal{N}_k}$ is acceptable in order to minimize the original objective function; in other words, the satisfaction of the SD condition (3) *would suggest that the point $\overline{x}_k$ provides a similar behaviour, regardless of the chosen mini-batch $\mathcal{N}_k$ or $\mathcal{D}_k$. In view of $\sum_{k=0}^{\infty} \zeta_k < \infty$, we have $\zeta_k \to 0$, so that the condition* (3) *becomes stricter as k increases. Furthermore, we highlight that there are no conditions on the size of $\mathcal{D}_k$, i.e., $\mathcal{D}_k$ can consist of only one element.*
We note that both decrease conditions (4) *and* (3) *are non-monotone.*

A crucial point of the behaviour of the proposed method is that the SD condition (3) cannot fail to be satisfied infinitely many times. In fact, if for many iterations this arises, as the size of the mini-batch is increased, there exists an iteration $\overline{k}$ such that, for $k \geq \overline{k}$, $N_k = N$ and the method is switched to a standard gradient method combined with a BB-like rule and the non-monotone line-search, for which there are well-known convergence results (see for example [12, 16, 13] and references therein). In this case, the algorithm is very expensive. Nevertheless, the experiments in Section 5 indicate that the SD condition (3) is satisfied in a vaste majority of cases and the number of discarded candidate points is very low and the size N for the mini-batch is never reached within the time equivalent to require the computation of 30 full gradients and to obtain a satisfactory accuracy.

3 Convergence analysis of LSNM-BB method for the finite sum case

Before we prove the main convergence result for the finite sum problem (2), we need to state the following two important lemmas. The proofs of these lemmas are fundamentally the same as the ones for Lemmas 1-2 in [5]. However, they are adapted to the **LSNM-BB** method and stated here for completeness.

Let us denote by $\mathcal{D}_k^+$ the subset of all possible outcomes of $\mathcal{D}_k$ at iteration k for which the condition (3) is satisfied, i.e.,

$$\mathcal{D}_k^+ = \{\mathcal{D}_k \subset \mathcal{N} \mid f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k\}. \quad (6)$$

We denote the complementary subset of outcomes at iteration k by

$$\mathcal{D}_k^- = \{\mathcal{D}_k \subset \mathcal{N} \mid f_{\mathcal{D}_k}(\overline{x}_k) > f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k\}. \quad (7)$$

The first lemma guarantees that if the mini-batches are always proper subsets of $\mathcal{N}$, then from a certain iteration forward the SD condition is always satisfied.

Lemma 1. *Suppose that Assumptions 1 and 2 hold. If $N_k < N$ for all $k \in \mathbb{N}$, then a. s. there exists $k_1 \in \mathbb{N}$ such that $\mathcal{D}_k^- = \emptyset$ for all $k \geq k_1$.*

Proof. Assume that $N_k < N$ for all $k \in \mathbb{N}$. Since the sample size sequence $\{N_k\}$ in **LSNM-BB** Algorithm is non-decreasing, this means that there exists some $\overline{N} < N$ and $k_2 \in \mathbb{N}$ such that $N_k = \overline{N}$ for all $k \geq k_2$. Now, let us assume that there is no $k_1 \in \mathbb{N}$ such that $\mathcal{D}_k^- = \emptyset$ for all $k \geq k_1$. This means that there exists an infinite sub-sequence of iterations $K \subseteq \mathbb{N}$ such that $\mathcal{D}_k^- \neq \emptyset$ for all $k \in K$. Since $\mathcal{D}_k$ is chosen randomly and uniformly, with finitely many possible

outcomes for each k, there exists some $q > 0$ such that $\mathbb{P}(\mathcal{D}_k \in \mathcal{D}_k^-) \geq q$ for all $k \in K$. So, we have

$$\mathbb{P}(\mathcal{D}_k \in \mathcal{D}_k^+, k \in K) \leq \Pi_{k \in K}(1-q) = 0;$$

this means that we will almost surely encounter an iteration at which the sample size will be increased due to violation of SD condition (3). This is a contradiction with the sample size being kept to $\overline{N}$ during the whole optimization process. Thus, we conclude that the statement holds. □

Next, we show that Lemma 1 implies that the Armijo-like inequality holds for the overall objective function for all k sufficiently large in the mini-batch scenario. The proof is essentially the same as the proof of Lemma 2 in [5], but we state it here for completeness.

Lemma 2. *Suppose that Assumptions 1 and 2 hold and that $\mathcal{D}_k$ is chosen with replacement. If $N_k < N$ for all $k \in \mathbb{N}$, then a. s.*

$$f_{\mathcal{N}}(\overline{x}_k) \leq f_{\mathcal{N}}(x_k) - c_{min}\|\nabla f_N(x_k)\|^2 + C_{max}\zeta_k$$

holds for all $k \geq k_1$ where k_1 is as in Lemma 1.

Proof. First, notice that Lemma 1 implies that a. s.

$$f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k \tag{8}$$

holds for all possible realizations of $\mathcal{D}_k$ and for all $k \geq k_1$. Thus, we conclude that a. s. for every $i = 1, 2, ..., N$ and every $k \geq k_1$ we have

$$F_i(\overline{x}_k) \leq F_i(x_k) - c_{min}\|\nabla F_i(x_k)\|^2 + C_{max}\zeta_k. \tag{9}$$

Indeed, if there exists $i \in \mathcal{N}$ that violates the previous inequality, then there would exist at least one realization of $\mathcal{D}_k$ (namely, $\mathcal{D}_k = \{i, i, ..., i\}$) that violates (8). Thus, a. s. for all $k \geq k_1$ we have

$$\begin{aligned} f_{\mathcal{N}}(\overline{x}_k) &= \frac{1}{N}\sum_{i=1}^{N} F_i(x_k) \leq \frac{1}{N}\sum_{i=1}^{N}(F_i(x_k) - c_{min}\|\nabla F_i(x_k)\|^2 + C_{max}\zeta_k \quad (10)\\ &= f_{\mathcal{N}}(x_k) - c_{min}\frac{1}{N}\sum_{i=1}^{N}\|\nabla F_i(x_k)\|^2 + C_{max}\zeta_k \\ &\leq f_{\mathcal{N}}(x_k) - c_{min}\|\nabla f_{\mathcal{N}}(x_k)\|^2 + C_{max}\zeta_k, \end{aligned}$$

where the last inequality comes from the fact that $\|\cdot\|^2$ is convex and therefore

$$\|\nabla f_{\mathcal{N}}(x_k)\|^2 = \|\frac{1}{N}\sum_{i=1}^{N}\nabla F_i(x_k)\|^2 \leq \frac{1}{N}\sum_{i=1}^{N}\|\nabla F_i(x_k)\|^2.$$

□

Next, we prove that the iterates of the proposed algorithm remain within a random level set.

Lemma 3. *Suppose that Assumptions 1 and 2 hold and that $\mathcal{D}_k$ is chosen with replacement. Then a. s. there exists a finite, random iteration $\tilde{k}$ such that*

$$f_{\mathcal{N}}(x_{\tilde{k}+k}) \leq f_{\mathcal{N}}(x_{\tilde{k}}) + \max\{1, C_{max}\}\bar{\zeta}$$

holds for all $k \in \mathbb{N}$.

Proof. If the full sample is reached, then there exists some $k_0 \in \mathbb{N}$ such that $N_k = N$ for all $k \geq k_0$ and (4) holds for a suitable t_k, i.e., for all $k \geq k_0$

$$f_{\mathcal{N}}(x_{k+1}) \leq f_{\mathcal{N}}(x_k) + \eta t_k g_k^T d_k + \zeta_k \leq f_{\mathcal{N}}(x_k) + \zeta_k.$$

Using the su mmability of ζ_k we obtain the result with $\tilde{k} = k_0$.

On the other hand, if $N_k < N$ for all k, then Lemma 1 implies the existence of k_1 such that $x_{k+1} = \bar{x}_k$ for all $k \geq k_1$ and Lemma 2 implies that

$$f_{\mathcal{N}}(x_{k+1}) \leq f_{\mathcal{N}}(x_k) - c_{min}\|\nabla f_{\mathcal{N}}(x_k)\|^2 + C_{max}\zeta_k \leq f_{\mathcal{N}}(x_k) + C_{max}\zeta_k$$

holds for all $k \geq k_1$ a. s. Again, using the su mmability of ζ_k we obtain the result with $\tilde{k} = k_1$. □

In order to prove the main convergence results for **LSNM-BB** Algorithm, we state the following assumption.

Assumption 3. *There exists a constant C such that $\mathbb{E}(|f_{\mathcal{N}}(x_{\tilde{k}})|) \leq C$, where $\tilde{k}$ is as in Lemma 3.*

The expectation in the previous assumption is taken over all possible sample paths. Assumption 3, together with the result of Lemma 3, implies that the sequence $\{f_{\mathcal{N}}(x_k)\}_{k \geq \tilde{k}}$ is uniformly bounded in expectation. Moreover, we obtain (see [5] for more details)

$$\mathbb{E}(|f_{\mathcal{N}}(x_{\tilde{k}})| \mid A) \leq C_1 \quad \text{and} \quad \mathbb{E}(|f_{\mathcal{N}}(x_{\tilde{k}})| \mid \bar{A}) \leq C_2, \tag{11}$$

where A represents a subset of all possible outcomes (sample paths) such that the full sample is reached eventually, $\bar{A}$ represents a subset of all possible outcomes which belong to the mini-batch scenario, and C_1, C_2 are some positive constants depending on C and the probability of the mini-batch scenario. Notice that Assumption 3 holds if we have bounded iterates and continuous objective function. Let us abbreviate $\mathbb{E}_A(\cdot) := \mathbb{E}(\cdot|A)$ and $\mathbb{E}_{\bar{A}}(\cdot) := \mathbb{E}(\cdot|\bar{A})$. We denote the corresponding conditional probabilities with $\mathbb{P}_A$ and $\mathbb{P}_{\bar{A}}$.

Now, we are ready to prove the main convergence result. The second part of the proof (the mini-batch scenario) is similar to the proof of Theorem 3.9 in [10], although in a different context.

Theorem 1. *Suppose that the Assumptions 1, 2 and 3 hold and let $\{x_k\}$ be a sequence generated by **LSNM-BB** Algorithm. Then*

$$\lim_{k \to \infty} \|\nabla f_{\mathcal{N}}(x_k)\| = 0 \quad a.s. \tag{12}$$

and each limit point of $\{x_k\}$ is stationary for problem (2) a.s.

Proof. Let us observe the first (mini-batch) scenario, where $N_k < N$ for all $k \in \mathbb{N}$. Then, Lemma 1 and 2 imply that a. s. there exists $k_1 \in \mathbb{N}$ such that the following holds for all $k \geq k_1$

$$f_{\mathcal{N}}(x_{k+1}) \leq f_{\mathcal{N}}(x_k) - c_{min}\|\nabla f_{\mathcal{N}}(x_k)\|^2 + C_{max}\zeta_k.$$

Equivalently, a. s. for all $s \in \mathbb{N}$ we have

$$f_{\mathcal{N}}(x_{k_1+s}) \leq f_{\mathcal{N}}(x_{k_1}) - c_{min}\sum_{\ell=0}^{s-1}\|\nabla f_{\mathcal{N}}(x_{k_1+\ell})\|^2 + C_{max}\sum_{\ell=0}^{s-1}\zeta_{k_1+\ell}.$$

Furthermore, applying the expectation $\mathbb{E}_{\bar{A}}$ and using the fact that $\tilde{k}$ coincides with k_1 in the mini-batch scenario, by (11) we obtain

$$\mathbb{E}_{\bar{A}}(f_{\mathcal{N}}(x_{\tilde{k}+s})) \leq C_2 - c_{min}\sum_{\ell=0}^{s-1}\mathbb{E}_{\bar{A}}(\|\nabla f_{\mathcal{N}}(x_{\tilde{k}+\ell})\|^2) + C_{max}\sum_{\ell=0}^{s-1}\zeta_{\tilde{k}+\ell}.$$

Moreover, using Assumptions 1 and 2 and letting $s \to \infty$ we obtain

$$\sum_{k=0}^{\infty}\mathbb{E}_{\bar{A}}(\|\nabla f_{\mathcal{N}}(x_k)\|^2) < \infty.$$

Now, by the extended version of Markov's inequality we have that for any $\epsilon > 0$

$$\mathbb{P}_{\bar{A}}(\|\nabla f_{\mathcal{N}}(x_k)\| \geq \epsilon) \leq \frac{\mathbb{E}_{\bar{A}}(\|\nabla f_{\mathcal{N}}(x_k)\|^2)}{\epsilon^2} < \infty$$

and therefore

$$\sum_{k=0}^{\infty}\mathbb{P}_{\bar{A}}(\|\nabla f_{N}(x_k)\| \geq \epsilon) < \infty.$$

Finally, Borel-Cantelli Lemma [15] implies that, conditioned on A, $\lim_{k\to\infty}\|\nabla f_{\mathcal{N}}(x_k)\| = 0$ a.s., or in other words

$$P(\lim_{i\to\infty}\|d(x_{\tilde{k}_1+i})\| = 0 \mid \bar{A}) = 1. \tag{13}$$

Now, let us consider the scenario where the full sample size is reached, i.e., the outcomes that belong to A. In this scenario, the method eventually becomes a standard gradient method equipped with BB step size and non-monotone backtracking line-search, but with a random "starting" point $\tilde{k} = k_0$. Notice that under the Assumption 1, both γ_k and t_k are uniformly bounded away from zero. Therefore, according to (4), we obtain the following inequality for all $k \geq \tilde{k}$

$$f_{\mathcal{N}}(x_{k+1}) \leq f_{\mathcal{N}}(x_k) - \eta\bar{t}\gamma_{min}\|\nabla f_{\mathcal{N}}(x_k)\|^2 + \zeta_k. \tag{14}$$

Applying the conditional expectation $\mathbb{E}_A$ and following similar steps as in the previous part of the proof, we obtain

$$P(\lim_{k\to\infty}\|\nabla f_{\mathcal{N}}(x_k)\| = 0 \mid A) = 1, \tag{15}$$

which together with (13) implies (12).
Finally, since we have proved that the gradient of the objective function tends to zero a.s. in all possible scenarios, by the continuity of $\nabla f_{\mathcal{N}}$ we conclude that every limit point of $\{x_k\}$ is stationary for $f_{\mathcal{N}}$ a.s. □

If $f_{\mathcal{N}}$ is $\mu_{\mathcal{N}}$-strongly convex we have a stronger convergence result. In this case, problem (2) has a unique solution x_* and for any $x \in \mathbb{R}^d$ the following inequality holds

$$\frac{\mu_{\mathcal{N}}}{2}\|x - x_*\|^2 \leq f_{\mathcal{N}}(x) - f_{\mathcal{N}}(x_*) \leq \frac{1}{2\mu_{\mathcal{N}}}\|\nabla f_{\mathcal{N}}(x)\|^2. \tag{16}$$

Taking x_k instead of x and letting $k \to \infty$, according to Theorem 1 we obtain the following result.

Corollary 1. *Let Assumptions 1, 2 and 3 hold and let $\{x_k\}$ be a sequence generated by **LSNM-BB** Algorithm. If $f_{\mathcal{N}}$ is $\mu_{\mathcal{N}}-$strongly convex, then the sequence $\{x_k\}$ converges a.s. to the unique solution x_* of problem (2).*

Next, we analyse the convergence rate and the worst-case complexity of the proposed algorithm under the strong convexity assumption. We show that R-linear convergence rate can be achieved. Moreover, if the local cost functions are heterogeneous enough, we provide an expected number of iterations to reach the ε-vicinity of the solution.

Theorem 2. *Suppose that the assumptions of Corollary 1 hold. Assume that $c_{min} < 1/(2\mu_{\mathcal{N}})$ and $\eta < 1/(2\mu_{\mathcal{N}}\gamma_{min}t_{min})$. Then the **LSNM-BB** Algorithm converges to the unique solution x_* of the problem (2) R-linearly in a mean squared sense, i.e., there exist constants $\rho \in (0,1)$ and $M > 0$ such that*

$$\mathbb{E}(\|x_{\tilde{k}+j} - x_*\|^2) \leq M\rho^j, \quad j = 1, 2, ... \tag{17}$$

where where $\tilde{k}$ is as in Lemma 3.

Proof. Following the steps of Theorem 1, in the mini-batch scenario we obtain that the following holds a. s. for all $k \geq k_1$

$$f_{\mathcal{N}}(x_{k+1}) \leq f_{\mathcal{N}}(x_k) - c_{min}\|\nabla f_{\mathcal{N}}(x_k)\|^2 + C_{max}\zeta_k.$$

Recall that in this scenario, k_1 coincides with $\tilde{k}$ from Lemma 3.

Now, by subtracting $f_{\mathcal{N}}(x_*)$ from both sides of the previous inequality and using the second inequality of (16) we obtain

$$f_{\mathcal{N}}(x_{k+1}) - f_{\mathcal{N}}(x_*) \leq \rho_1(f_{\mathcal{N}}(x_k) - f_{\mathcal{N}}(x_*)) + C_{max}\zeta_k,$$

where $\rho_1 = 1 - 2c_{min}\mu_{\mathcal{N}} \in (0,1)$. Therefore, a. s. for each $j \in \mathbb{N}$ there holds

$$f_{\mathcal{N}}(x_{k_1+j}) - f_{\mathcal{N}}(x_*) \leq \rho_1^j(f_{\mathcal{N}}(x_{k_1}) - f_{\mathcal{N}}(x_*)) + s_{k_1+j},$$

where $s_{k_1+j} = C_{max}\sum_{i=1}^{j}\rho_1^{i-1}\zeta_{k_1+j-i}$. Since the sequence of ζ_k is assumed to be nonnegative and summable, we conclude that it converges to zero R-linearly. Furthermore, without loss of generality, we can assume that the sequence of ζ_k is monotone decreasing, so

$$s_{k_1+j} \leq C_{max}\sum_{i=1}^{j}\rho_1^{i-1}\zeta_{j-i} = s_j.$$

Moreover, the sequence of s_j also converges to zero R-linearly (see Lemma 4.2 in [23] for instance), so there exist constants $M_\zeta > 0$ and $\rho_\zeta \in (0,1)$ such that for each $j \in \mathbb{N}$

$$f_{\mathcal{N}}(x_{k_1+j}) - f_{\mathcal{N}}(x_*) \leq \rho_1^j(f_{\mathcal{N}}(x_{k_1}) - f_{\mathcal{N}}(x_*)) + M_\zeta\rho_\zeta^j.$$

Applying the conditional expectation $\mathbb{E}_{\bar{A}}$ we obtain

$$\mathbb{E}_{\bar{A}}(f_{\mathcal{N}}(x_{k_1+j}) - f_{\mathcal{N}}(x_*)) \leq \rho_1^j(C_2 - f_{\mathcal{N}}(x_*)) + M_\zeta \rho_\zeta^j$$

and conclude the existence of constants $M_{\bar{A}} > 0$ and $\rho_{\bar{A}} \in (0,1)$ such that for each $j \in \mathbb{N}$

$$\mathbb{E}_{\bar{A}}(f_{\mathcal{N}}(x_{k_1+j}) - f_{\mathcal{N}}(x_*)) \leq M_{\bar{A}} \rho_{\bar{A}}^j. \tag{18}$$

Analogously, for the full sample scenario we obtain the existence of constants $M_A > 0$ and $\rho_A \in (0,1)$ such that for each $j \in \mathbb{N}$

$$\mathbb{E}_A(f_{\mathcal{N}}(x_{k_0+j}) - f_{\mathcal{N}}(x_*)) \leq M_A \rho_A^j, \tag{19}$$

where k_0 coincides with $\tilde{k}$ and the analysis is based on (14) and the assumption $\eta < 1/(2\mu_{\mathcal{N}}\gamma_{min}t_{min})$.
Finally, combining both scenarios we obtain

$$\begin{aligned} \mathbb{E}(f_{\mathcal{N}}(x_{\tilde{k}+j}) - f_{\mathcal{N}}(x_*)) &= P(\bar{A})\mathbb{E}_{\bar{A}}(f_{\mathcal{N}}(x_{\tilde{k}}) - f_{\mathcal{N}}(x_*)) + P(A)\mathbb{E}_A(f_{\mathcal{N}}(x_{\tilde{k}}) - f_{\mathcal{N}}(x_*)) \\ &= P(\bar{A})\mathbb{E}_{\bar{A}}(f_{\mathcal{N}}(x_{k_1}) - f_{\mathcal{N}}(x_*)) + P(A)\mathbb{E}_A(f_{\mathcal{N}}(x_{k_0}) - f_{\mathcal{N}}(x_*)) \\ &\leq \rho^j \bar{M}, \end{aligned} \tag{20}$$

where $\rho = \max\{\rho_A, \rho_{\bar{A}}\}$ and $\bar{M} = \max\{M_A, M_{\bar{A}}\}$. Applying the strong convexity assumption we conclude the proof of the statement with $M = \frac{2\bar{M}}{\mu_{\mathcal{N}}}$. □

Corollary 2. *Let Assumptions of Theorem 2 hold. Then* $\mathbb{E}(\|x_k - x_*\|^2) \leq \varepsilon$ *holds for all* $k \geq \hat{k}$*, where*

$$\hat{k} = \tilde{k} + \left\lceil \frac{|\log(\varepsilon/M)|}{|\log(\rho)|} \right\rceil.$$

and $\tilde{k}$ *is like in Lemma 3.*

Notice that the worst-case complexity in the previous corollary is of order $\mathcal{O}(\log(\varepsilon))$, but it also depends on random iteration $\tilde{k}$. In general, $\tilde{k}$ is very hard to determine since it is problem-dependent. In the sequel, we provide an estimate for $\tilde{k}$ under assumption of heterogeneous local cost functions. More precisely, we assume that for each k there exists at least one F_i function that violates the inequality (9). We formalize this assumption as follows.

Assumption 4. *For each* k *there exists at least one* F_i *function such that*

$$F_i(\overline{x}_k) > F_i(x_k) - c_{min}\|\nabla F_i(x_k)\|^2 + C_{max}\zeta_k.$$

Assumption 4 implies that $\mathcal{D}_k^-$ is nonempty for each k and there exists $p \in (0,1]$ such that

$$P(\mathcal{D}_k \in \mathcal{D}_k^-) \geq p > 0. \tag{21}$$

For instance, if $|\mathcal{D}_k| = 1$, then $p \geq 1/N$. Furthermore, let S_k be a random variable that counts how many times the sample size is increased within the first k iterations. We can represent S_k as a sum of indicator variables, i.e., $S_k = I_1 + ... + I_k$, where $I_k = 1$ if $N_k > N_{k-1}$ and $I_k = 0$ otherwise. Notice that $\mathbb{E}(I_k) = P(I_k = 1) = P(\mathcal{D}_k \in \mathcal{D}_k^-) \geq p$ and thus

$$\mathbb{E}(S_k) \geq kp. \tag{22}$$

Furthermore, notice that the full sample size is reached a.s. under the Assumption 4. Let us denote by $\tilde{N}$ the number of increments of sample size needed to reach the full sample. For instance, if in line 22 of LSNM-BB algorithm we set $N_{k+1} = N_k + 1$, then $\tilde{N} = N - N_0$. If we use $N_{k+1} = \lceil \theta N_k \rceil$, then $\tilde{N}$ is at most $\lceil \log(N/N_0)/\log(\theta) \rceil$. Therefore, by setting $\mathbb{E}(S_{\tilde{k}}) = \tilde{N}$ and using (22) we conclude that the expected number of iterations to reach the full sample is bounded from above by $\lceil \tilde{N}/p \rceil$. We summarize this analysis in the following statement.

Corollary 3. *Let Assumptions of Theorem 2 hold together with Assumption 4. Then the expected number of iterations to reach* $\mathbb{E}(\|x_k - x_*\|^2) \leq \varepsilon$ *is*

$$\hat{k}_E = \lceil \tilde{N}/p \rceil + \left\lceil \frac{|\log(\varepsilon/M)|}{|\log(\rho)|} \right\rceil.$$

4 The LSNM-BB-G method for the general case and its convergence analysis

Now, let us consider more general problem (1). We assume that the approximation $f_{\mathcal{N}_k}(x)$ of the objective function at iteration k is formed by the SAA estimator

$$f_{\mathcal{N}_k}(x) = \frac{1}{N_k} \sum_{i=1}^{N_k} F(x, \xi_i^k), \tag{23}$$

where $\xi_1^k, \xi_2^k, ..., \xi_{N_k}^k$ are i.i.d. random vectors for each k and $N_k = |\mathcal{N}_k|$. We assume the same structure for the additional sampling, i.e.,

$$f_{\mathcal{D}_k}(x) = \frac{1}{D_k} \sum_{i=1}^{D_k} F(x, \tilde{\xi}_i^k), \tag{24}$$

where $\tilde{\xi}_1^k, ..., \tilde{\xi}_{D_k}^k$ are i.i.d. random vectors independent of $\xi_1^k, \xi_2^k, ..., \xi_{N_k}^k$, and $D_k = |\mathcal{D}_k|$. We also assume that, given a point x, both $f_{\mathcal{N}_k}(x)$ and $f_{\mathcal{D}_k}(x)$ are unbiased estimators of $f(x)$. The SAA estimator with unbounded sample size is a key point in the generalization of the method **LSNM-BB**. The modified version, named **LSNM-BB-G** and shown in Algorithm 2, implements the details of the proposed method for solving the problem (1). A further difference between the two algorithms is that the sample size D_k used for the estimator $f_{\mathcal{D}_k}$ increases at the same time as the sample size N_k used for the estimator $f_{\mathcal{N}_k}$, but not necessary at the same rate and there holds $D_k \leq N_k$.

Notice that the proposed algorithm can yield two possible scenarios regarding the sample size: 1) the mini-batch scenario where the sample size N_k is bounded from above as well as D_k; 2) the scenario where both N_k and D_k tend to infinity. The second scenario corresponds to the case where the trial point is rejected infinitely many times. We assume that the sequence of iterates is bounded. Moreover, we also make an assumption that, given a point x, the gradient $\nabla F(x, \xi)$ is a.s. bounded with $D(x)$. This ensures that the stochastic gradient is a.s. well defined at any given point x. Moreover, it is satisfied in logistic regression problems if the attributes belong to some finite range.

Algorithm 2 LSNM-BB-G

Given $x_0 \in \mathbb{R}^d$, η, $\beta \in (0,1)$, $\{\zeta_k\} \subset \mathbb{R}_+$ subject to $\sum_{k=0}^{\infty} \zeta_k \leq \overline{\zeta} < \infty$, c_{min}, $C_{max} \in \mathbb{R}_{++}$, $0 < \gamma_{min} < \gamma_{max}$, $\gamma_0 \in [\gamma_{min}, \gamma_{max}]$, $N_0 > 0$, $\theta \geq \tilde{\theta} > 1$.
Set $k \leftarrow 0$, $j \leftarrow 0$
Choose i.i.d. random sample $\xi_1^k, \xi_2^k, ..., \xi_{N_0}^k$ to form $f_{\mathcal{N}_0}$ by (23)
Compute $g_0 \leftarrow \nabla f_{\mathcal{N}_0}(x_0)$
while the stopping criterion is not satisfied **do**
    Compute $m(N_j)$
    $i = 1$
    repeat
        Compute $d_k \leftarrow -\gamma_k g_k$
        Find the smallest integer $\ell \geq 0$ such that $t_k = \beta^\ell$ satisfies

$$f_{\mathcal{N}_k}(x_k + t_k d_k) \leq f_{\mathcal{N}_k}(x_k) + \eta t_k g_k^T d_k + \zeta_k \tag{25}$$

        $\overline{x}_k \leftarrow x_k + t_k d_k$
        if $N_k < N$ **then**
            Choose i.i.d. random sample $\tilde{\xi}_1^k, \tilde{\xi}_2^k, ..., \tilde{\xi}_{D_k}^k$ to form $f_{\mathcal{D}_k}$ by (24)
            if $f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k$ **then**
                $x_{k+1} \leftarrow \overline{x}_k$
                $\mathcal{N}_{k+1} \leftarrow \mathcal{N}_k$
                $N_{k+1} \leftarrow N_k$
                Compute $g_{k+1} \leftarrow \nabla f_{\mathcal{N}_{k+1}}(x_{k+1})$
                Compute γ_{k+1} by a BB-like rule with threshold in $[\gamma_{min}, \gamma_{max}]$
            else
                $x_{k+1} \leftarrow x_k$
                Choose $N_{k+1} > N_k$
                Choose $D_{k+1} \in (D_k, N_{k+1}]$
                $k \leftarrow k + 1$
                Exit and go to step 35
            end if
        else $(\mathcal{N}_k = \mathcal{N})$
            $x_{k+1} \leftarrow \overline{x}_k$
            Compute $g_{k+1} \leftarrow \nabla f_{\mathcal{N}}(x_{k+1})$
            Compute γ_{k+1} by a BB-like rule with threshold in $[\gamma_{min}, \gamma_{max}]$
        end if
        $k \leftarrow k + 1$
        $i = i + 1$
    until $i > m(N_j)$ OR the stopping criterion is satisfied
        if $N_k < N$ **then**
            Choose i.i.d. random sample $\xi_1^k, \xi_2^k, ..., \xi_{N_k}^k$ to form $f_{\mathcal{N}_k}$ by (23)
            Compute $g_k \leftarrow \nabla f_{N_k}(x_k)$
            Compute γ_k as $\frac{1}{\|g_k\|}$ with threshold in $[\gamma_{min}, \gamma_{max}]$
        end if
        $j \leftarrow k$
    end while

Assumption 5. *The function $F(\cdot, \xi)$ is bounded from below and continuously-differentiable with L-Lipschitz continuous gradient for any given ξ. Moreover, for every x there exists a constant $D(x)$ such that $\|\nabla F(x, \xi)\| \leq D(x)$ for almost every ξ.*

Consequently, all the SAA functions ($f_{\mathcal{N}_k}$ and $f_{\mathcal{D}_k}$) are bounded from below and continuously-differentiable with L-Lipschitz continuous gradients as well.

Assumption 6. *The sequence of iterates $\{x_k\}_{k\in\mathbb{N}}$ generated by Algorithm LSNM-BB-G is bounded.*

In order to continue with the convergence analysis, let us denote by $e(x, \mathcal{N})$ the error of the SAA estimate based on sample $\mathcal{N}$ at point x, i.e.,

$$e(x, \mathcal{N}) := |f_{\mathcal{N}}(x) - f(x)|. \tag{26}$$

We also define the corresponding error for the gradient as

$$e_g(x, \mathcal{N}) := |\|\nabla f_{\mathcal{N}}(x)\|^2 - \|\nabla f(x)\|^2|. \tag{27}$$

In order to claim a.s. convergence of these errors, we make the following assumption.

Assumption 7. *The function F and its gradient ∇F are dominated by integrable functions on any compact subset of $\mathbb{R}^d$.*

Since we assume i.i.d. samples, the Uniform Law of Large Numbers (ULLN) implies that, under Assumption 7, there holds

$$\lim_{|\mathcal{N}|\to\infty} \sup_{x\in S} e(x, \mathcal{N}) = 0 \text{ a.s. and } \lim_{|\mathcal{N}|\to\infty} \sup_{x\in S} e_g(x, \mathcal{N}) = 0 \text{ a.s.}, \tag{28}$$

for any given compact set S (see Theorems 7.48 and 7.52 from [4]).

Now, we state the conditions under which we have infinite number of iterations at which the trial point is accepted unless we have encountered a stationary point of the objective function f.

Lemma 4. *Suppose that the Assumptions 2, 5, 6 and 7 hold. Then, a.s., there exists an infinite subset of iterations $K \subseteq \mathbb{N}$ such that $x_{k+1} = \overline{x}_k$ for each $k \in K$ provided that $\|\nabla f(x_k)\| > 0$ for each k and $c_{min} < \eta\gamma_{min}\min\{2\beta(1-\eta)/L, 1\}$.*

Proof. First, notice that $t_k \geq \min\{2\beta(1-\eta)/L, 1\}$ due to the Assumption 5 and the backtracking line-search. Therefore, from (25) we obtain that

$$f_{\mathcal{N}_k}(\overline{x}_k) \leq f_{\mathcal{N}_k}(x_k) - \eta t_k\gamma_k\|\nabla f_{\mathcal{N}_k}(x_k)\|^2 + \zeta_k \leq f_{\mathcal{N}_k}(x_k) - \underline{c}\|\nabla f_{\mathcal{N}_k}(x_k)\|^2 + \zeta_k, \tag{29}$$

where $\underline{c} := \eta\gamma_{min}\min\{2\beta(1-\eta)/L, 1\}$. Furthermore, using (26) and (27) we obtain

$$f(\overline{x}_k) \leq f(x_k) - \underline{c}\|\nabla f(x_k)\|^2 + e(\overline{x}_k, \mathcal{N}_k) + e(x_k, \mathcal{N}_k) + \underline{c}e_g(x_k, \mathcal{N}_k) + \zeta_k. \tag{30}$$

Now, let us assume that $K \subseteq \mathbb{N}$ such that $x_{k+1} = \overline{x}_k$ for each $k \in K$ does not exist. This means that there exists $\overline{k}$ such that the point $\overline{x}_k$ is rejected for all

$k \geq \overline{k}$, i.e., the sequence of iterates becomes stationary at $x_{\overline{k}}$. According to the algorithm, this is possible only if the following happens for all $k \geq \overline{k}$

$$f_{\mathcal{D}_k}(\overline{x}_k) > f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k.$$

Thus, for all $k \geq \overline{k}$ we have

$$\begin{aligned} f(\overline{x}_k) >& f(x_k) - c_{min}\|\nabla f(x_k)\|^2 - e(\overline{x}_k, \mathcal{D}_k) - e(x_k, \mathcal{D}_k) + \\ & - c_{min} e_g(x_k, \mathcal{D}_k) + C_{max}\zeta_k. \end{aligned} \tag{31}$$

We observe that $C_{max}\zeta_k \geq 0$ in the previous inequality; then, combining (30) and (31) and using the fact that $x_k = x_{\overline{k}}$ for all $k \geq \overline{k}$ we obtain

$$\begin{aligned} (\underline{c} - c_{min})\|\nabla f(x_{\overline{k}})\|^2 <& e(\overline{x}_k, \mathcal{D}_k) + e(x_{\overline{k}}, \mathcal{D}_k) + c_{min} e_g(x_{\overline{k}}, \mathcal{D}_k) \\ & + e(\overline{x}_k, \mathcal{N}_k) + e(x_{\overline{k}}, \mathcal{N}_k) + \underline{c}\, e_g(x_{\overline{k}}, \mathcal{N}_k) + \zeta_k. \end{aligned}$$

Notice that each rejection of a trial point increases both N_k and D_k, so in this scenario we have that $N_k \to \infty$ and $D_k \to \infty$ and thus

$$\lim_{k\to\infty} e(x_{\overline{k}}, \mathcal{D}_k) + c_{min} e_g(x_{\overline{k}}, \mathcal{D}_k) + e(x_{\overline{k}}, \mathcal{N}_k) + \underline{c}\, e_g(x_{\overline{k}}, \mathcal{N}_k) = 0 \ \text{ a.s..}$$

Moreover, Assumption 5 implies the existence of a constant $D(x_{\overline{k}})$ such that $\|\nabla f_{\mathcal{N}_k}(x_{\overline{k}})\| \leq D(x_{\overline{k}})$ a.s. which further implies that

$$\|\overline{x}_k\| \leq \|x_{\overline{k}}\| + t_k\gamma_k\|\nabla f_{\mathcal{N}_k}(x_{\overline{k}})\| \leq \|x_{\overline{k}}\| + \gamma_{max} D(x_{\overline{k}}) \quad \text{a.s. for all } k \geq \overline{k}.$$

Thus, we conclude that for all $k \geq \overline{k}$, $\overline{x}_k$ remains bounded, i.e., $\{\overline{x}_k\}_{k\geq\overline{k}} \in \mathcal{B}(x_{\overline{k}}, \gamma_{max} D(x_{\overline{k}}))$ a.s. and applying the ULLN we obtain

$$\lim_{k\to\infty} e(\overline{x}_k, \mathcal{N}_k) + e(\overline{x}_k, \mathcal{D}_k) = 0 \ \text{ a.s.}$$

Since ζ_k is summable, we have $\lim_{k\to\infty} \zeta_k = 0$ and thus, by using the assumption that $c_{min} < \underline{c}$ and $\|\nabla f(x_{\overline{k}})\| > 0$, we obtain the following contradiction

$$0 < (\underline{c} - c_{min})\|\nabla f(x_{\overline{k}})\|^2 \leq 0 \ \text{ a.s.}$$

This completes the proof. □

The following lemma is the analogue of Lemma 1, stated in the previous section for the finite sum case.

Lemma 5. *Suppose that the Assumptions 2 and 5 hold. If $N_k \leq \overline{N} < \infty$ for all $k \in \mathbb{N}$ then there exists $k_1 \in \mathbb{N}$ such that the SD condition* (3) *of* ***LSNM-BB-G*** *Algorithm holds a.s. for all $k \geq k_1$.*

Proof. Assume that there exists some $k_2 \in \mathbb{N}$ such that $N_k = \overline{N}$ for all $k \geq k_2$. Now, let us assume that there is no $k_1 \in \mathbb{N}$ such that the SD condition (3) holds a.s. for all $k \geq k_1$. This means that there exists an infinite sub-sequence of iterations $K \subseteq \mathbb{N}$ such that the measure of $\mathcal{D}_k^-$ is non-zero for all $k \in K$ and thus the probability of $\mathcal{D}_k \in \mathcal{D}_k^-$ is strictly positive for all $k \in K$. This further implies that $\mathbb{P}(\mathcal{D}_k \in \mathcal{D}_k^+) < 1$ for all $k \in K$ and

$$\mathbb{P}(\mathcal{D}_k \in \mathcal{D}_k^+, k \in K) = \prod_{k\in K} \mathbb{P}(\mathcal{D}_k \in \mathcal{D}_k^+) = 0.$$

This means that we will almost surely encounter an iteration at which the sample size will be increased due to violation of SD condition (3). This is a contradiction with the sample size being kept to $\overline{N}$ during the optimization process. Thus, we conclude that the statement holds. □

Now, we prove the main convergence result for **LSNM-BB-G** Algorithm.

Theorem 3. *Suppose that the assumptions of Lemma 4 hold and let* $\{x_k\}$ *be a sequence generated by* ***LSNM-BB-G*** *Algorithm. Then, we have*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0 \quad a.s. \tag{32}$$

and there exists a limit point of $\{x_k\}$ *which is stationary for problem* (1) *a.s.*

Proof. Recall that there are two possible scenarios for the proposed algorithm: mini-batch scenario ($\bar{A}$) and the one where the sample size tends to infinity (A).

Let us observe the first scenario. In this case, Lemma 5 implies that the following holds a.s. for all $k \geq k_1$

$$f_{\mathcal{D}_k}(\overline{x}_k) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k. \tag{33}$$

This further implies that $\bar{x}_k = x_{k+1}$ a.s. for all $k \geq k_1$. Now, let us denote by $\mathcal{F}_{k+1/2}$ the σ-algebra generated by $\mathcal{N}_0$, $\mathcal{D}_0$, ..., $\mathcal{N}_{k-1}$, $\mathcal{D}_{k-1}$, $\mathcal{N}_k$. Since $\mathcal{D}_k$ is chosen randomly and uniformly after both $\overline{x}_k$ and x_k are determined, we have

$$f(\bar{x}_k) = \mathbb{E}(f_{\mathcal{D}_k}(\overline{x}_k) \mid \mathcal{F}_{k+1/2}) \quad \text{and} \quad f(x_k) = \mathbb{E}(f_{\mathcal{D}_k}(x_k) \mid \mathcal{F}_{k+1/2}).$$

Thus, applying conditional expectation with respect to $\mathcal{F}_{k+1/2}$ on (33) leads to

$$f(\bar{x}_k) \leq f(x_k) - c_{min}\mathbb{E}(\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 \mid \mathcal{F}_{k+1/2}) + C_{max}\zeta_k. \tag{34}$$

Moreover, there holds

$$\mathbb{E}(\nabla f_{\mathcal{D}_k}(x_k)|\ \mathcal{F}_{k+1/2}) = \nabla f(x_k), \tag{35}$$

so we obtain

$$\begin{aligned}\|\nabla f(x_k)\|^2 = \|\mathbb{E}(\nabla f_{\mathcal{D}_k}(x_k)|\ \mathcal{F}_{k+1/2})\|^2 &\leq \mathbb{E}^2(\|\nabla f_{\mathcal{D}_k}(x_k)\||\ \mathcal{F}_{k+1/2}) \\ &\leq \mathbb{E}(\|\nabla f_{\mathcal{D}_k}(x_k)\|^2|\ \mathcal{F}_{k+1/2}).\end{aligned}$$

and conclude that for all $k \geq k_1$ there holds

$$f(\bar{x}_k) \leq f(x_k) - c_{min}\|\nabla f(x_k)\|^2 + C_{max}\zeta_k. \tag{36}$$

Furthermore, since we also have $\bar{x}_k = x_{k+1}$ a.s. for all $k \geq k_1$ and f is a deterministic function, we know that for all $k \geq k_1$ there holds

$$\mathbb{E}_{\bar{A}}(f(x_{k+1})) = \mathbb{E}_{\bar{A}}(f(\bar{x}_k)) \leq \mathbb{E}_{\bar{A}}(f(x_k)) - c_{min}\mathbb{E}_{\bar{A}}(\|\nabla f(x_k)\|^2) + C_{max}\zeta_k. \tag{37}$$

Now, using the Assumptions 2 and 6, we conclude that $\sum_{k=k_1}^{\infty} \mathbb{E}_{\bar{A}}(\|\nabla f(x_k)\|^2) < \infty$ and continuing as in the proof of Theorem 1, we conclude

$$P(\lim_{k\to\infty} \|\nabla f(x_k)\| = 0 \mid \bar{A}) = 1. \tag{38}$$

Now, let us observe the second scenario, where N_k tends to infinity. Recall that in this scenario D_k tends to infinity as well. Lemma 4 implies the existence of an infinite sub-sequence of iterations $K := \{k_j\}_{j\in\mathbb{N}} \subseteq \mathbb{N}$ such that the trial point is accepted. In other words, for each $k \in K$ we have

$$f_{\mathcal{D}_k}(x_{k+1}) \leq f_{\mathcal{D}_k}(x_k) - c_{min}\|\nabla f_{\mathcal{D}_k}(x_k)\|^2 + C_{max}\zeta_k$$

and by using (26) and (27) we get

$$f(x_{k+1}) \leq f(x_k) - c_{min}\|\nabla f(x_k)\|^2 + a_k,$$

where

$$a_k := e(x_{k+1}, \mathcal{D}_k) + e(x_k, \mathcal{D}_k) + c_{min}e_g(x_k, \mathcal{D}_k) + C_{max}\zeta_k. \tag{39}$$

In other words, for all $j \in \mathbb{N}$ we have

$$f(x_{k_j+1}) \leq f(x_{k_j}) - c_{min}\|\nabla f(x_{k_j})\|^2 + a_{k_j}.$$

Moreover, in all the intermediate iterations between k_j and k_{j+1}, the trial point is rejected and we conclude that $x_{k_j+1} = ... = x_{k_{j+1}-1} = x_{k_{j+1}}$ for each j and thus

$$f(x_{k_{j+1}}) = ... = f(x_{k_j+1}) \leq f(x_{k_j}) - c_{min}\|\nabla f(x_{k_j})\|^2 + a_{k_j}. \tag{40}$$

Now, assume that (32) does not hold. This means that there exists $\varepsilon > 0$ such that $\|\nabla f(x_k)\|^2 \geq \varepsilon$ for each $k \in \mathbb{N}$. Thus, for each j we obtain

$$f(x_{k_{j+1}}) \leq f(x_{k_j}) - c_{min}\varepsilon + a_{k_j}.$$

Due to Assumption 6, summability of ζ_k and the fact that $D_k \to \infty$, we conclude that $\lim_{j\to\infty} a_{k_j} = 0$ a.s.; thus, there exists $\overline{j}$ such that $a_{k_j} \leq c_{min}\varepsilon/2$ for all $j \geq \overline{j}$ a.s. which implies that the following holds

$$f(x_{k_{j+1}}) \leq f(x_{k_j}) - c_{min}\varepsilon/2.$$

This further implies that, for any $s \in \mathbb{N}$, a.s. there holds

$$f(x_{k_{\overline{j}+s}}) \leq f(x_{k_{\overline{j}}}) - sc_{min}\varepsilon/2,$$

and by employing Assumption 6 and letting $s \to \infty$, we obtain $\lim_{s\to\infty} f(x_{k_{\overline{j}+s}}) = -\infty$, which is a contradiction with f being bounded from below. Thus, we conclude that (32) must hold, i.e.,

$$P(\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0 \mid A) = 1. \tag{41}$$

Thus, combining (38) and (41) we conclude that (32) holds.
Finally, since we assume that the sequence of iterates is bounded, due to the fact that f is continuously differentiable function, we conclude that a.s. there exists an accumulation point of the sequence $\{x_k\}$ which is stationary for function f. □

Remark 3. *Notice that we can prove a stronger results (* $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ *a.s.) if the sequence of* a_k *defined in* (39) *is summable. This can be achieved for some classes of problems if we impose fast enough increase of* D_k *in the scenario where* $D_k \to \infty$ *(see the discussion after the proof of Theorem 3.1 in [2] and the references therein.)*

If the function f is μ-strongly convex, then we can prove that the whole sequence converges to the unique solution x_* of problem (1).

Theorem 4. *Suppose that the Assumptions of Lemma 4 hold. Further, assume that the function f is μ-strongly convex and $c_{min} < 1/(2\mu)$. Then the sequence $\{x_k\}$ generated by **LSNM-BB-G** Algorithm converges to x_* a.s.*

Proof. Notice all the assumptions of Theorem 3 hold. Moreover, we have proved that in the mini-batch scenario we obtain $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ a.s. and the statement is a direct consequence of the following inequality

$$\frac{\mu}{2}\|x_k - x_*\|^2 \leq f(x_k) - f(x_*) \leq \frac{1}{2\mu}\|\nabla f(x_k)\|^2. \tag{42}$$

In the second scenario, when $D_k \to \infty$, there holds (40), i.e.,

$$f(x_{k_{j+1}}) \leq f(x_{k_j}) - c_{min}\|\nabla f(x_{k_j})\|^2 + a_{k_j},\ j \in \mathbb{N}$$

where $K := \{k_j\}_{j\in\mathbb{N}} \subseteq \mathbb{N}$ is the sequence as in the proof of Theorem 3 - an infinite sub-sequence of iterations at which the trial point is accepted, and $a_{k_j} \to 0, j \to \infty$ a.s.. Now, by using the second inequality in (42) and subtracting $f(x_*)$ from both sides of the previous inequality, we obtain

$$f(x_{k_{j+1}}) - f(x_*) \leq f(x_{k_j}) - f(x_*) - c_{min}2\mu(f(x_{k_j}) - f(x_*)) + a_{k_j},$$

i.e., for each j there holds

$$f(x_{k_{j+1}}) - f(x_*) \leq \theta(f(x_{k_j}) - f(x_*)) + a_{k_j},$$

where $\theta := 1 - c_{min}2\mu \in (0,1)$ by the assumption on c_{min}. Applying the induction argument we obtain

$$f(x_{k_j}) - f(x_*) \leq \theta^j(f(x_{k_0}) - f(x_*)) + \sum_{t=1}^{j} \theta^{j-t} a_{k_{t-1}}.$$

Since $f(x_{k_0}) - f(x_*)$ is bounded, we have $\lim_{j\to\infty} \theta^j(f(x_{k_0}) - f(x_*)) = 0$. Moreover, since a_{k_j} tends to zero a.s. as j tends to infinity, it can be shown that $\lim_{j\to\infty} \sum_{t=1}^{j} \theta^{j-t} a_{k_{t-1}} = 0$ a.s. (see Lemma 3.1. in [3] for instance). Therefore, we conclude that $\lim_{j\to\infty} f(x_{k_j}) = f(x_*)$ a.s., and according to (42) we have that $\lim_{j\to\infty} \|x_{k_j} - x_*\| = 0$ a.s. Having in mind the definition of the subsequence $\{x_{k_j}\}_{j\in\mathbb{N}}$, we conclude that the whole sequence of iterates $\{x_k\}_{k\in\mathbb{N}}$ tends to x_* a.s. □

Under the assumptions of the previous theorem, we analyze the convergence rate of the proposed method. We prove R-linear convergence in a mean squared sense if the sample size is increased fast enough in the scenario A ($N_k \to \infty$), i.e., if the following holds for all j and some constants $M_a > 0$ and $\rho_a \in (0,1)$

$$\mathbb{E}_A(a_{k_j}) \leq M_a \rho_a^j, \tag{43}$$

where a_{k_j} is as in the proof of Theorem 4. We refer to iterations where $x_{k+1} = \bar{x}_k$ as successful iterations.

Theorem 5. *Let assumptions of Theorem 4 hold. Then the sequence of successful iterations generated by* ***LSNM-BB-G*** *Algorithm tends to x_* R-linearly in the mean squared sense, provided that* (43) *holds.*

Proof. In the mini-batch scenario ($\bar{A}$) there holds (37). Following the analysis of the proof of Theorem 2, we conclude that for all $j \in \mathbb{N}$

$$\mathbb{E}_{\bar{A}}(\|x_{k_1+j} - x_*\|^2) \leq \frac{2M_{\bar{A}}}{\mu}\rho_{\bar{A}}^j.$$

Notice that all the iterations after k_1 are a.s. successful. Considering the remaining scenario (A), we conclude that (40) holds. By applying the conditional expectation $\mathbb{E}_A$, subtracting $f(x_*)$ from both sides, using (43) and strong convexity assumption, we obtain that the following holds for all successful iterations, i.e., for all $j \in \mathbb{N}$

$$\mathbb{E}_A(f(x_{k_{j+1}}) - f(x_*)) \leq \rho_1 \mathbb{E}_A(f(x_{k_j}) - f(x_*)) + M_a\rho_a^j,$$

where $\rho_1 = (1 - c_{min}2\mu) \in (0, 1)$. Furthermore, using the same arguments as in the proof of Theorem 2, we conclude that

$$\mathbb{E}_A(\|x_{k_j} - x_*\|^2) \leq \tilde{\rho}_A^j \tilde{M}_A,$$

for some $\tilde{\rho}_A \in (0, 1)$ and $\tilde{M}_A > 0$. This completes the proof. □

We conclude the analysis by stating the worst-case complexity result that takes into account dissimilarity of approximate functions given by (21).

Corollary 4. *Let assumptions of Theorem 5 hold together with* (21)*. Then the expected number of iterations to reach $\mathbb{E}(\|x_k - x_*\|^2) \leq \varepsilon$ is*

$$\tilde{k}_E = \left\lceil \frac{|\log(\varepsilon/\tilde{M}_A)|}{|(1-p)\log(\tilde{\rho}_A)|} \right\rceil.$$

Proof. Notice that under Assumption 3 and its consequence (21), the number of unsuccessful iterations is infinite a.s. Indeed, the probability of having finitely many unsuccessful iterations is 0 since

$$P(\mathcal{D}_k \in \mathcal{D}_k^+, k \geq k_2) = \prod_{k \geq k_2} (1 - P(\mathcal{D}_k \in \mathcal{D}_k^-)) \leq \prod_{k \geq k_2} (1 - p) = 0.$$

Thus, the sample size tends to infinity, i.e., the scenario A happens a.s. Thus, according to the proof of the previous theorem, we have $\mathbb{E}_A(\|x_{k_j} - x_*\|^2) \leq \tilde{\rho}_A^j \tilde{M}_A$, for all j and $\mathbb{E}(\|x_{k_j} - x_*\|^2) \leq \varepsilon$ is satisfied for all $j \geq \tilde{\tilde{j}}$ where

$$\tilde{\tilde{j}} = \left\lceil \frac{|\log(\varepsilon/\tilde{M}_A)|}{|\log(\tilde{\rho}_A)|} \right\rceil.$$

Recall that k_j represent successful iterations and therefore this means that the ε-vicinity of the solution in the mean squares sense is attained after at most $\tilde{\tilde{j}}$ successful iterations. Notice also that we can set $x_{k_0} = x_0$ since the first successful iteration must be in the initial point (but not necessarily at initial

iteration $k = 0$). Furthermore, by using (22), we conclude that the expected number of successful iterations within total k iterations is

$$\mathbb{E}(k - S_k) \leq k - kp.$$

Finally, setting the expected number of successful iterations to reach $\tilde{j}$, i.e., $\mathbb{E}(k - S_k) \geq \tilde{j}$, we obtain the result. □

5 Numerical experiments

In this section we evaluate the numerical effectiveness of the proposed approach when we have to solve a binary classification problem, i.e., a minimization problem as (2). Only in the last Section 5.4 we consider a problem that simulates the formulation of the problem (1). We remark that, in order to prevent possible overfitting problems, an ℓ_2 regularization term is included in the objective function in both cases.
In the numerical experiments addressing problem (2), we consider four datasets: *w8a*, *IJCNN* and *RCV1* (downloadable from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/`) and *MNIST* (available at `https://yann.lecun.com/exdb/mnist/`). This last dataset is adapted to the binary classification problem, by separating the items in even and the odd digits. The details of the considered datasets are reported in Table 1. Here, d is the size of any element ξ_i (feature vector and corresponding label) of the dataset $\mathcal{N}$ (training set) and of the testing set.

dataset	$d-1$	#training set (N)	#testing set
MNIST	784	60000	10000
w8a	300	44774	4975
IJCNN	22	49990	91701
RCV1	47236	20242	10000

Table 1: Dataset features.

We consider two different binary classifiers, corresponding to two loss functions, one convex and one non-convex. In particular, by denoting as $a_i \in \mathbb{R}^{d-1}$ and $b_i \in \{1, -1\}$ the feature vector and the class label of the i-th example respectively, in the first case the objective function in (2) is the sum of N convex logistic regression (LR) loss terms penalized by an ℓ_2 regularization term:

$$F(x, \xi_i) = \log\left[1 + e^{-b_i a_i^T x}\right] + \lambda \|x\|^2,$$

whereas, in the second case, the objective function is the sum of N non-convex loss in 2-layer neural networks (NN) penalized by an ℓ_2 regularization term:

$$F(x, \xi_i) = \left(1 - \frac{1}{1 + e^{-b_i a_i^T x}}\right)^2 + \lambda \|x\|^2.$$

Here, λ is the regularization parameter. For all test problems, we set $\lambda = 10^{-4}$. All numerical results described in the following (unless otherwise specified) were

obtained by running each numerical test 10 times with the same setting, but leaving the possibility to the random number generator to vary; consequently, each reported performance measurement is the average of 10 values.
Furthermore, for any considered test problem, a ground truth value f^* for the minimum of the objective function was computed by a huge number of iterations of the standard stochastic gradient method (SGD).
In order to evaluate the results carried out from the numerical experiments, the following averaged quantities are tracked:

- averaged optimality gap value at each epoch, where the optimality gap value is defined as $|f_{\mathcal{N}}(x) - f^*|$ and it is computed by using the training set; the *epoch* is a measure of computational complexity, equivalent to the computation of a full gradient;
- averaged mini-batch size at each iteration;
- averaged accuracy evaluated on the testing set at each epoch.

As a further performance measure, we can use also the decreasing rate of the objective function at the iteration k, defined as

$$R_k = \frac{f_{\mathcal{N}}(x_k) - f^*}{f_{\mathcal{N}}(x_0) - f^*},$$

where $f_{\mathcal{N}}(x_k)$ is the averaged value of the objective function computed at the current k-th iterate over 10 runs.
Finally, in all the numerical experiments, the methods are stopped when the total number of the objective function gradient evaluations exceeds $N \cdot maxit$ where $maxit$ is the number of considered epochs. This stopping criterion is motivated by the choice to fix the computational complexity (i.e., the available budget of the computational resources), although in view of the stochasticity of the methods, the number of the iterations at any run may be different.
In all numerical experiments described below (unless otherwise indicated), 30 epochs were considered for the running of **LSNM-BB** Algorithm (or other methods), that is, we fixed a budget of computational resources equivalent to computing 30 full gradients of the objective function. In addition, we remark that, for both the considered objective functions, the evaluation of each term at the current iterate x_k provides also the value of the related gradient at x_k with negligible additional computational costs.

5.1 Performance evaluation of LSNM-BB with respect to the rule for $m(N_j)$

In order to evaluate the performance of **LSNM-BB** Algorithm we consider the ABB_{min} rule for the updating of γ_k within each j-th cycle. This choice is motivated by a numerical comparison between different BB rules, confirming that ABB_{min} rule enables to obtain the best performance, as it is well known in the literature [6]. The other hyperparameters involved in **LSNM-BB** Algorithm do not influence significantly the effectiveness of the method, as the numerical experimentation reported in the Section 5.2 will highlight; thus, in all numerical tests performed, we use the same setting, i.e., $C_{max} = 1$, $c_{min} = 10^{-4}$, $\eta = 10^{-4}$, $\beta = 10^{-2}$, $\zeta_k = 0.99^k$, $k = 0, 1, \ldots$, the size of the initial mini-batch

$N_0 = 5$, $\gamma_{min} = 10^{-8}$, $\gamma_{max} = 10^8$; furthermore, the increasing rule for the mini-batch size N_k at step 22 (when the trial update $\overline{x}_k$ is refused) is set as $N_{k+1} \leftarrow \min(N_k + 1, N)$.
The first experiment is aimed to evaluate the behaviour of **LSNM-BB** Algorithm with respect to the maximum number $m(N_j)$ of the steps of the inner cycle (steps 8-33). We observe that the strategy of using cycles of iterations, where a cycle is related to the same mini-batch $\mathcal{N}_k$, is not required by the theoretical convergence analysis of the method. Indeed this is a feature of the practical implementation of the method, aimed at exploiting the well-known effectiveness of the BB-like rules. Motivated by this approach, we evaluate the behaviour of different rules in the definition of $m(N_j)$:

Case 1) $m(N_j) = 10$;

Case 2) $m(N_j) = \lfloor \sqrt{N_j} \rfloor$;

Case 3) $m(N_j) = \lfloor 0.2 N_j \rfloor$;

Case 4) $m(N_j) = \max(\lfloor \log(N_j) \rfloor, 1)$.

		Case 1)		*Case 2)*		*Case 3)*		*Case 4)*	
		LR	NN	LR	NN	LR	NN	LR	NN
MNIST	$R_{\overline{K}}$	$1.49{\cdot}10^{-3}$	$7.35{\cdot}10^{-3}$	$1.33{\cdot}10^{-3}$	$8.85{\cdot}10^{-3}$	$1.53{\cdot}10^{-3}$	$1.06{\cdot}10^{-2}$	$1.19{\cdot}10^{-3}$	$1.05{\cdot}10^{-2}$
	$N_{\overline{K}} \pm$ STD	889 ± 32.77	828± 60.44	899± 18.31	828± 28.85	893± 14.79	760± 50.95	887± 21.62	822± 35.19
	$E_{\overline{K}} \pm$ STD	39 %± 1.11 %	29 %± 1.39 %	39 %± 0.58 %	28 %± 0.66 %	39 %± 0.59 %	28 %± 1.18 %	40 %± 0.87 %	28 %± 0.82 %
w8a	$R_{\overline{K}}$	$8.00{\cdot}10^{-4}$	$1.60{\cdot}10^{-3}$	$1.08{\cdot}10^{-3}$	$1.69{\cdot}10^{-3}$	$8.69{\cdot}10^{-4}$	$1.45{\cdot}10^{-3}$	$1.19{\cdot}10^{-3}$	$1.33{\cdot}10^{-3}$
	$N_{\overline{K}} \pm$ STD	940± 26.70	934± 30.88	955± 21.35	923± 54.11	942± 23.99	923± 49.72	932± 31.29	925± 30.25
	$E_{\overline{K}} \pm$ STD	39 %± 0.92 %	38 %± 0.81 %	38 %± 0.61 %	37 %± 1.46 %	38 %± 0.72 %	35 %± 1.44 %	40 %± 0.89 %	38 %± 0.85 %
IJCNN	$R_{\overline{K}}$	$4.46{\cdot}10^{-3}$	$3.32{\cdot}10^{-3}$	$4.83{\cdot}10^{-3}$	$3.19{\cdot}10^{-3}$	$3.51{\cdot}10^{-3}$	$3.94{\cdot}10^{-3}$	$5.86{\cdot}10^{-3}$	$3.59{\cdot}10^{-3}$
	$N_{\overline{K}} \pm$ STD	976± 33.57	962± 43.94	967± 39.42	959± 20.53	972± 27.12	975± 35.87	955± 30.24	971± 28.67
	$E_{\overline{K}} \pm$ STD	40 %± 1.08 %	36 %± 1.17 %	40 %± 1.21 %	37 %± 0.49 %	40 %± 0.94 %	36 %± 0.93 %	40 %± 1.17 %	37 %± 0.93 %
RCV1	$R_{\overline{K}}$	$5.86{\cdot}10^{-3}$	$4.04{\cdot}10^{-3}$	$6.23{\cdot}10^{-3}$	$2.99{\cdot}10^{-3}$	$4.88{\cdot}10^{-3}$	$3.36{\cdot}10^{-3}$	$4.08{\cdot}10^{-3}$	$1.98{\cdot}10^{-3}$
	$N_{\overline{K}} \pm$ STD	590± 21.32	572± 31.82	597± 15.92	607± 19.29	583± 26.82	609± 19.18	597± 20.48	611± 13.98
	$E_{\overline{K}} \pm$ STD	33 %± 0.93 %	32 %± 1.14 %	32 %± 0.68 %	32 %± 0.84 %	31 %± 0.96 %	31 %± 0.72 %	32 %± 0.93 %	32 %± 0.54 %

Table 2: **LSNM-BB** Algorithm: comparison of different rules for the definitions of $m(N_j)$; LR and NN denote the cases of convex and non-convex loss terms in the objective function respectively.

Table 2 summarizes the results obtained by applying **LSNM-BB** Algorithm to both the two objective functions for all the datasets; in particular, for any combination of dataset-objective function-choice of $m(N_j)$, we report:

- the value $R_{\overline{K}}$ of the objective function decreasing rate at the final iteration $\overline{K}$;
- the averaged mini-batch size at the final iteration $\overline{K}$ with the related standard deviation;
- the averaged percentage $E_{\overline{K}}$ of the early exits in the inner cycles, i.e., the ratio between the number of times when the $m(N_j)$ iterations of the inner cycle fail to complete (and an early exit occurs) and the total number of iterations $\overline{K}$; the standard deviation is reported.

Table 2 highlights that the values of $R_{\overline{K}}$ are very similar in all cases (around 10^{-3}). Also the percentage of the early exits is contained within 40%. The final mini-batch size is limited with almost the same values for all rules. Consequently, the method appears very robust with respect to the choice of the rule for $m(N_j)$.
This analysis can be further explored by examining the results obtained for the

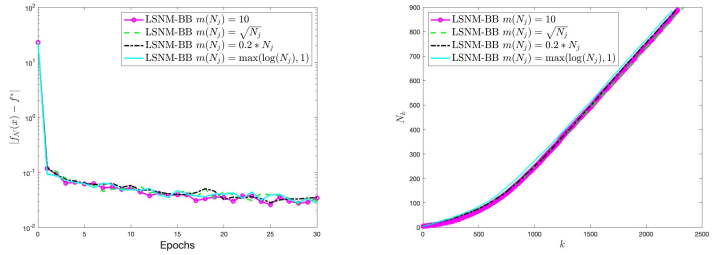

Figure 1: Results obtained by **LSNM-BB** Algorithm for the dataset *MNIST* and objective function with LR loss terms - On the left, behaviour of the averaged optimality gap with respect to the epochs in the case of the four rules for $m(N_j)$; on the right, corresponding behaviour of the averaged mini-batch size with respect to the iterations.

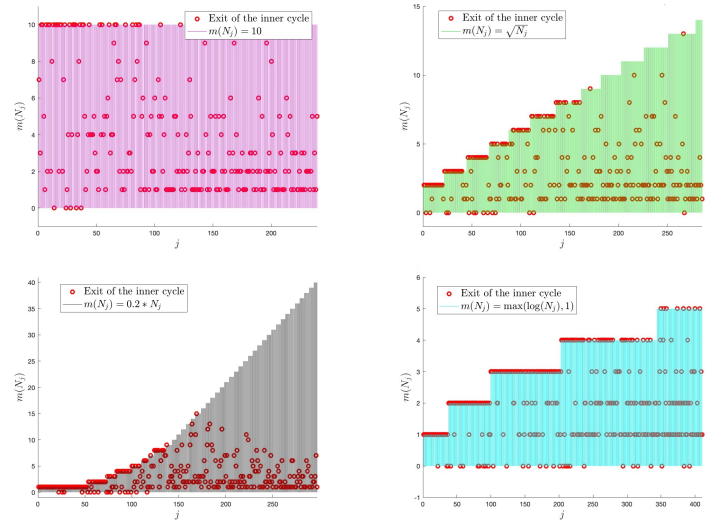

Figure 2: Results obtained by **LSNM-BB** Algorithm for the dataset *MNIST* and objective function with LR loss terms - Depictions of the running of the inner cycle for the four rules of $m(N_j)$ over the first 2 epochs: top-left plot $m(N_j) = 10$, top-right plot $m(N_j) = \sqrt{N_j}$; bottom-left plot $m(N_j) = 0.2\ N_j$, bottom-right $m(N_j) = \max(\log(N_j), 1)$. The red dot in a position different from $(j, m(N_j))$ means that the SD condition is not satisfied by the trial iterate and the inner cycle early exits.

MNIST database in more detail. Figures 1 and 2 concern the numerical behaviour of **LSNM-BB** when the objective function includes LR loss terms for the four different choices of $m(N_j)$. In particular, Figure 1 shows the averaged optimality gap with respect to the epochs (on the left) and the averaged increase of the mini-batch size with respect to the iterations (on the right); Figure 2 depicts the running of the inner cycle for the four rules of $m(N_j)$ over the first 2 epochs: in each plot, for any j-th inner cycle (depicted on the horizontal axis), the corresponding expected number of iterations $m(N_j)$ is drawn on the vertical axis, together with the information (marked with the red dot) about the itera-

tion at which the cycle stops. If the red dot is at the point $(j, m(N_j))$, the whole cycle is executed, while, on the contrary, an early exit occurs, highlighting that the SD condition (3) is not satisfied by the current trial iterate.
From Figures 1 and 2 no significant difference depending on $m(N_j)$ can be observed in the effectiveness of **LSNM-BB** Algorithm, both regarding the optimality gap and the increase in the mini-batch size. In Figure 2, we can detect that, in the first two epochs, the rules that determine a rapid increase of $m(N_j)$, as $\sqrt{N_j}$ or $0.2\ N_j$, determine the early exit of the inner cycle almost always from a certain iteration; this does not happen for $m(N_j) = 10$ or $\max(\log(N_j), 1)$. So we deduce that the rule for the inner cycle shouldn't increase too much.

A similar analysis was repeated for the binary classification of the dataset

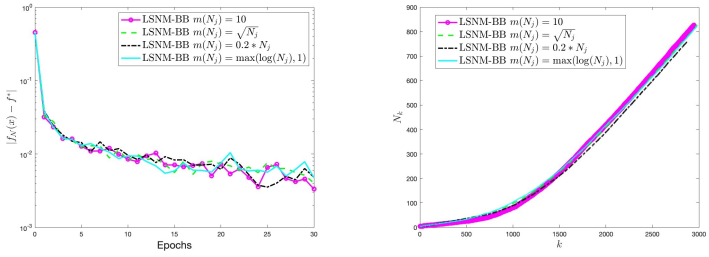

Figure 3: Results obtained by **LSNM-BB** Algorithm for the dataset *MNIST* and objective function with NN loss terms - On the left, behaviour of the averaged optimality gap with respect to the epochs in the case of the four rules for $m(N_j)$; on the right, corresponding behaviour of the averaged mini-batch size with respect to the iterations.

MNIST by minimizing the non-convex objective function with NN loss terms by **LSNM-BB** Algorithm. Figures 3-4 show the obtained results, very similar to those of the previous test problem. In conclusion, the effectiveness of **LSNM-BB** Algorithm appears not too dependent on the prefixed number of iterations of the inner cycle, since this number remains small. In the following numerical experiments, we decided to use the rule $m(N_j) = \max(\log(N_j), 1)$, since with this choice the increase in the number of internal iterations is slow.

5.2 About the stability of LSNM-BB Algorithm with respect to the initial setting

In this subsection, we describe the behaviour of the **LSNM-BB** Algorithm with respect to the initial setting of the hyperparameters involved in the SD condition (3), i.e. c_{min}, C_{max} and ζ_k. The behaviour of the values of η and β related to the standard line-search (4) has been deeply investigated in the deterministic framework; so that we choose standard values, as $\eta = 10^{-4}$ and $\beta = 10^{-2}$. Furthermore, the starting mini-batch size is set as a small value $N_0 = 5$, to prevent considering the entire dataset too prematurely; the bounds for the learning rate γ_{min} and γ_{max} are chosen within a very wide range of variability to evaluate the effectiveness of the BB-like rules.
In the following experiment, we consider six different configurations for the hyperparameters c_{min}, C_{max} and ζ_k, $k \in \mathcal{N}$:

1. $c_{min} = 10^{-4}$, $C_{max} = 1$, $\zeta_k = 0.99^k$;

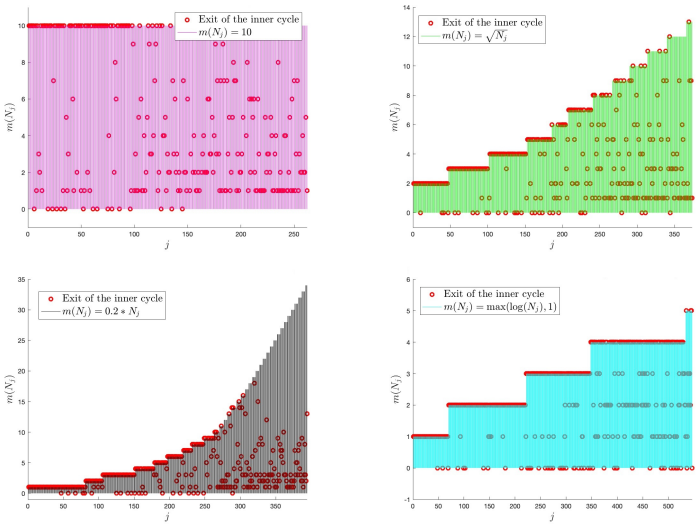

Figure 4: Results obtained by **LSNM-BB** Algorithm for the dataset *MNIST* and objective function with NN loss terms - Depictions of the running of the inner cycle for the four rules of $m(N_j)$ over the first 2 epochs: top-left plot $m(N_j) = 10$, top-right plot $m(N_j) = \sqrt{N_j}$; bottom-left plot $m(N_j) = 0.2\ N_j$, bottom-right $m(N_j) = \max(\log(N_j), 1)$. The red dot in a position different from $(j, m(N_j))$ means that the SD condition is not satisfied by the trial iterate and the inner cycle early exits.

2. $c_{min} = 10^{-4}$, $C_{max} = 10$, $\zeta_k = 0.99^k$;
3. $c_{min} = 10^{-4}$, $C_{max} = 0.1$, $\zeta_k = 0.99^k$;
4. $c_{min} = 10^{-4}$, $C_{max} = 1$, $\zeta_k = 0.1^k$;
5. $c_{min} = 10^{-2}$, $C_{max} = 1$, $\zeta_k = 0.99^k$;
6. $c_{min} = 10^{-1}$, $C_{max} = 10$, $\zeta_k = 0.99^k$.

In Table 3 we report the results obtained by running **LSNM-BB** Algorithm equipped with the six initial settings in the case of the objective functions with convex and non-convex terms for the four datasets. In particular, for any combination of dataset-objective function-initial setting, we report at the end of 30 epochs the following performance measures:

- the decreasing rate $R_{\overline{K}}$ of the considered objective function and the averaged mini-batch size at the final iteration $\overline{K}$ with the related standard deviation;
- the averaged percentage $E_{\overline{K}}$ of the early exits with the related standard deviation.

From the Table 3 we can observe that different starting settings do not qualitatively influence the results; indeed, in every configurations, $R_{\overline{K}}$ takes a value with the same order of magnitude for each test problem; the same

		MNIST		*w8a*		*IJCNN*		*RCV1*	
		LR	NN	LR	NN	LR	NN	LR	NN
Setting 1	$R_{\overline{K}}$	$1.20{\cdot}10^{-3}$	$1.04{\cdot}10^{-2}$	$8.47{\cdot}10^{-4}$	$1.33{\cdot}10^{-3}$	$5.86{\cdot}10^{-3}$	$3.59{\cdot}10^{-3}$	$4.09{\cdot}10^{-3}$	$1.98{\cdot}10^{-3}$
	$N_{\overline{K}}$ ± STD	887 ± 21.63	822± 35.19	932± 31.29	932± 30.25	955± 30.24	968± 28.67	597± 20.48	612± 13.98
	$E_{\overline{K}}$ ± STD	40 %± 0.87 %	28 %± 0.82 %	38 %± 0.89 %	38 %± 0.84 %	40 %± 1.17 %	37 %± 0.93 %	32 %± 0.93 %	32 %± 0.54 %
Setting 2	$R_{\overline{K}}$	$1.34{\cdot}10^{-3}$	$1.16{\cdot}10^{-2}$	$8.41{\cdot}10^{-4}$	$1.34{\cdot}10^{-3}$	$4.73{\cdot}10^{-3}$	$3.30{\cdot}10^{-3}$	$3.96{\cdot}10^{-3}$	$2.19{\cdot}10^{-3}$
	$N_{\overline{K}}$ ± STD	902± 14.71	789± 43.81	931± 26.66	958± 24.10	973± 22.83	975± 32.98	593± 12.58	610± 13.89
	$E_{\overline{K}}$ ± STD	36 %± 0.47 %	26 %± 0.94 %	34 %± 0.66 %	35 %± 0.69 %	36 %± 0.63 %	34 %± 0.84 %	29 %± 0.45 %	29 %± 0.55 %
Setting 3	$R_{\overline{K}}$	$1.49{\cdot}10^{-3}$	$9.84{\cdot}10^{-3}$	$8.27{\cdot}10^{-4}$	$1.80{\cdot}10^{-3}$	$4.52{\cdot}10^{-3}$	$3.51{\cdot}10^{-3}$	$5.61{\cdot}10^{-3}$	$2.81{\cdot}10^{-3}$
	$N_{\overline{K}}$± STD	878± 28.48	784± 38.72	907± 35.54	971± 16.67	943± 30.54	955± 35.08	583± 25.42	590± 16.00
	$E_{\overline{K}}$± STD	43 %± 1.23 %	31 %± 1.06 %	41 %± 1.33 %	42 %± 0.52 %	44 %± 1.18 %	42 %± 1.36 %	36 %± 1.32 %	36 %± 0.79 %
Setting 4	$R_{\overline{K}}$	$1.64{\cdot}10^{-3}$	$1.30{\cdot}10^{-2}$	$9.90{\cdot}10^{-4}$	$1.70{\cdot}10^{-3}$	$4.88{\cdot}10^{-3}$	$2.94{\cdot}10^{-3}$	$6.45{\cdot}10^{-3}$	$3.87{\cdot}10^{-3}$
	$N_{\overline{K}}$ ± STD	900± 23.13	862± 26.65	960± 20.67	994± 22.28	966± 35.82	990± 19.43	587± 25.79	596± 16.70
	$E_{\overline{K}}$ ± STD	48 %± 1.09 %	45 %± 1.39 %	56 %± 1.27 %	63 %± 1.29 %	51 %± 1.69 %	53 %± 1.02 %	42 %± 1.37 %	40 %± 0.95 %
Setting 5	$R_{\overline{K}}$	$1.93{\cdot}10^{-3}$	$1.34{\cdot}10^{-2}$	$8.04{\cdot}10^{-4}$	$1.52{\cdot}10^{-3}$	$4.33{\cdot}10^{-3}$	$3.46{\cdot}10^{-3}$	$3.82{\cdot}10^{-3}$	$1.85{\cdot}10^{-3}$
	$N_{\overline{K}}$ ± STD	985± 20.71	844± 47.43	928± 23.41	961± 28.19	975± 21.37	970± 31.30	615± 26.40	619± 19.80
	$E_{\overline{K}}$ ± STD	54 %± 0.95 %	33 %± 1.12 %	38 %± 0.75 %	39 %± 0.77 %	40 %± 0.74 %	38 %± 0.98 %	35 %± 1.22 %	34 %± 0.80 %
Setting 6	$R_{\overline{K}}$	$2.84{\cdot}10^{-3}$	$1.11{\cdot}10^{-2}$	$7.41{\cdot}10^{-4}$	$1.47{\cdot}10^{-3}$	$5.36{\cdot}10^{-3}$	$2.65{\cdot}10^{-3}$	$1.21{\cdot}10^{-3}$	$1.04{\cdot}10^{-3}$
	$N_{\overline{K}}$ ± STD	1026± 23.93	905± 35.27	958± 25.04	978± 22.34	990± 25.54	989± 20.27	690± 10.99	689± 10.38
	$E_{\overline{K}}$ ± STD	59 %± 0.99 %	36 %± 0.86 %	37 %± 0.67 %	38 %± 0.57 %	40 %± 0.76 %	37 %± 0.54 %	41 %± 0.37 %	38 %± 0.39 %

Table 3: **LSNM-BB** Algorithm: numerical results of the comparison with respect to six different settings of the hyperparameters C_{min}, C_{max} and ζ_k; LR and NN denote the cases of convex and non-convex terms in the objective function, respectively.

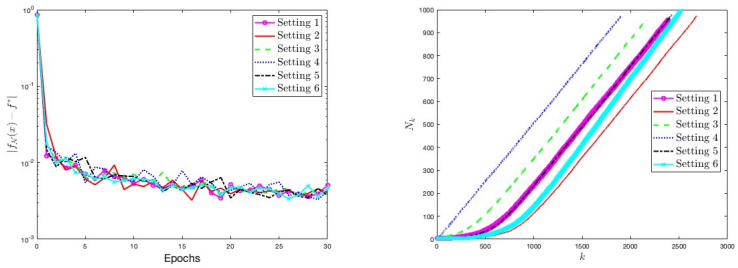

Figure 5: Results obtained by **LSNM-BB** Algorithm for the dataset *IJCNN* and the objective function with LR loss terms - On the left, behaviour of the averaged optimality gap computed at the training set with respect to the epochs for the six starting settings of c_{min}, C_{max} and ζ_k. On the right, increase of the averaged mini-batch size with respect to the iterations for **LSNM-BB** Algorithm equipped with the six starting settings of c_{min}, C_{max} and ζ_k.

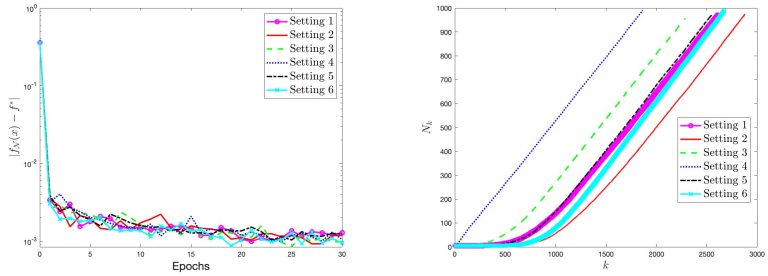

Figure 6: Results obtained by **LSNM-BB** Algorithm for the dataset *IJCNN* and the objective function with NN loss terms - On the left, behaviour of the averaged optimality gap computed at the training set with respect to the epochs for the six starting settings of c_{min}, C_{max} and ζ_k. On the right, increase of the averaged mini-batch size with respect to the iterations for **LSNM-BB** Algorithm equipped with the six starting settings of c_{min}, C_{max} and ζ_k.

consideration can be repeated also for the final mini-batch size and $E_{\overline{K}}$, whose values are in very narrow ranges for each column of the table.

To highlight the results of Table 3, Figures 5 and 6 show the behaviour of the averaged optimality gap (on the left) and the increase of the mini-batch size (on the right), obtained by running **LSNM-BB** Algorithm with the six different starting settings for the dataset *IJCNN* with objective function with LR and NN terms, respectively. We can observe that the behaviour of the averaged optimality gap appears to be unaffected by the values chosen for c_{min}, C_{max} and ζ_k. Similarly, the values of the final mini-batch size are really similar, although the number of iterations varies independently of a more or less severe SD condition.
However, for a fixed computational budget, the results obtained by **LSNM-BB** Algorithm appear not to depend significantly on the values chosen for c_{min}, C_{max} and ζ_k. For the next subsections we use **LSNM-BB** Algorithm with the first configuration.

5.3 Comparison of LSNM-BB Algorithm with several state of the art methods

The next experiment is aimed to compare **LSNM-BB** Algorithm with an effective version of the standard deterministic descent method and several state of the art methods in the stochastic framework. The methods taken into consideration to carry out a comparison are listed below with the implementation details:

- standard full gradient descent method **GD-BB** equipped with the ABB_{min} rule and the Armijo non-monotone line-search [6, 7]; using the notation in [6, 7], we set $\tau = 0.9$, the memory size for the BB2 definition is 2; the step length obtained by the ABB_{min} rule is thresholded within the range $[10^{-8}, 10^{8}]$;

- **SGD** method with the best tuned value of α_0 as starting learning rate and constant mini-batch size equal to 50; the value of α_0 is determined by a very expensive trial and error procedure; in order to guarantee the convergence, the value of the learning rate is decreased by a rule with behaviour as $\mathcal{O}(1/k)$ [8], i.e., $\alpha_k = \frac{\alpha_0}{\max(k/1200, nepoch)}$; where *nepoch* is the counter for the epochs;

- **SARAH** method with the setting of the hyperparameters as specified in [24];

- SGD method with momentum (**SGD-MOM**), where the stochastic gradient is computed at the iteration k as

$$g_k = (1 - \beta)\nabla f_{\mathcal{N}_k}(x_k) + \beta g_{k-1},$$

  with $g_0 = \nabla f_{\mathcal{N}_0}(x_0)$ and $\mathcal{N}_k$ a random sub-sample of fixed size; we set $\beta = 0.7$, 50 as mini-batch size and the best tuned value as fixed learning rate [1];

- Adaptive Sampling Method (**ASM**) [25], based on the increase in the mini-batch size depending on a test (Augmented Inner Test) which checks

whether the current stochastic gradient is a descent direction in expectation; at any iteration, the learning rate is computed by means a line-search procedure; the initial mini-batch size is set as $N_0 = 3$ whereas the initial learning rate is $\alpha_0 = 1$; using the same notation in [25], the setting of the other hyperparameters is $\theta = 0.9$, $\nu = 5.84$, $r = 5$, $\gamma = 0.38$, $\eta = 2$ and $\zeta_k = \zeta = 2$.

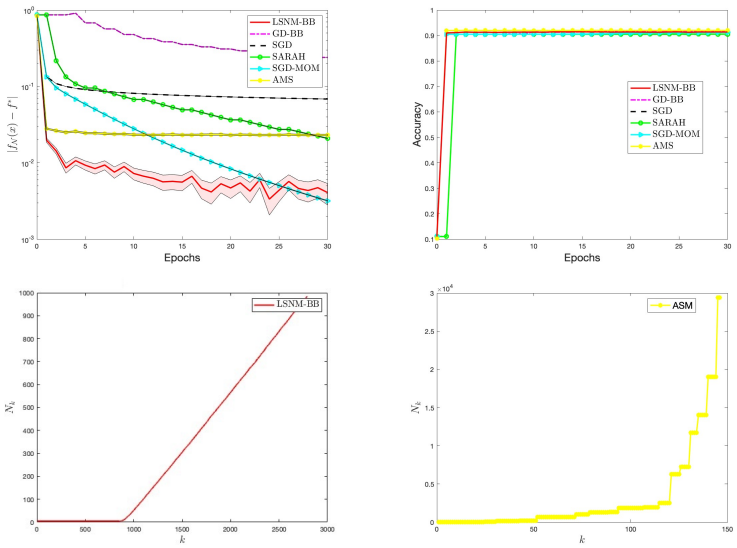

Figure 7: Dataset *IJCNN* and objective function with LR loss terms - First row: on the left, behaviour of the averaged optimality gap (combined with the relative standard deviation) computed at the training set with respect to the epochs; on the right, averaged accuracy evaluated at the testing set with respect to the epochs. Second row: increase of the averaged mini-batch size with respect to the iterations for **LSNM-BB** method (on the left) and for **ASM** method (on the right).

Dataset	*MNIST*			*w8a*			*IJCNN*			*RCV1*		
Method	$R_{\overline{K}}$	*Acc*	OG± STD	$R_{\overline{K}}$	*Acc*	OG± STD	$R_{\overline{K}}$	*Acc*	OG± STD	$R_{\overline{K}}$	*Acc*	OG± STD
LSNM-BB	$6.65{\cdot}10^{-4}$	0.8924	0.0153± 0.0079	$1.17{\cdot}10^{-3}$	0.9007	0.0064 ± 0.0027	$4.72{\cdot}10^{-3}$	0.9138	0.0041 ± 0.0013	$1.06{\cdot}10^{-2}$	0.9433	0.0340± 0.0074
GD-BB	$6.81{\cdot}10^{-3}$	0.8502	0.1565	$3.37{\cdot}10^{-1}$	0.8926	1.8536	$2.55{\cdot}10^{-1}$	0.9050	0.2204	$2.92{\cdot}10^{-12}$	0.9541	$0.93\cdot10^{-11}$
SGD	$2.33{\cdot}10^{-3}$	0.8806	0.0536± 0.0003	$5.99{\cdot}10^{-3}$	0.8890	0.0330 ± 0.0002	$7.92{\cdot}10^{-2}$	0.9050	0.0684 ± 0.0002	$8.58{\cdot}10^{-1}$	0.5013	2.7464± 0.0012
SARAH	$9.58{\cdot}10^{-4}$	0.8928	0.0220± 0.0010	$2.62{\cdot}10^{-3}$	0.8935	0.0144 ± 0.0005	$2.42{\cdot}10^{-2}$	0.9053	0.0209 ± 0.0001	$6.43{\cdot}10^{-1}$	0.5227	2.0597± 0.0000
SGD-MOM	$3.38{\cdot}10^{-4}$	0.8975	0.0078± 0.0003	$9.00{\cdot}10^{-4}$	0.8990	0.0050 ± 0.0001	$3.68{\cdot}10^{-3}$	0.9124	0.0032 ± 0.0000	$4.46{\cdot}10^{-1}$	0.7227	1.4278 ± 0.0003
ASM	$1.65{\cdot}10^{-3}$	0.8820	0.0382± 0.0016	$6.18{\cdot}10^{-4}$	0.8984	0.0035± 0.0015	$2.77{\cdot}10^{-2}$	0.9202	0.0232 ± 0.0006	$2.69{\cdot}10^{-2}$	0.9464	0.0878± 0.0051

Table 4: Comparison between different methods for objective function with LR loss terms; at the final iteration $\overline{K}$, the following performance measures are reported: $R_{\overline{K}}$ computed at the training set, the averaged accuracy *Acc* computed at the testing set and the averaged optimality gap at the training set ± the relative standard deviation (OG± STD).

In Figures 7 and 8 we can observe the behaviour in 30 epochs for all considered methods for the dataset *IJCNN* and the objective functions with LR and NN

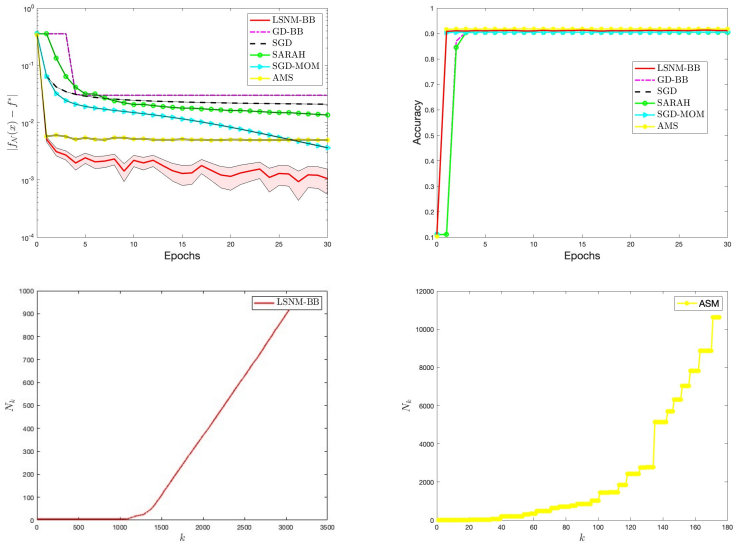

Figure 8: Dataset *IJCNN* and objective function with NN loss terms - First row: on the left, behaviour of the averaged optimality gap (combined with the relative standard deviation) computed at the training set with respect to the epochs; on the right, averaged accuracy evaluated at the testing set with respect to the epochs. Second row: increase of the averaged mini-batch size with respect to the iterations for **LSNM-BB** method (on the left) and for **ASM** method (on the right).

Dataset	*MNIST*			*w8a*			*IJCNN*			*RCV1*		
Method	$R_{\overline{K}}$	*Acc*	OG± STD	$R_{\overline{K}}$	*Acc*	OG± STD	$R_{\overline{K}}$	Acc	OG± STD	$R_{\overline{K}}$	*Acc*	OG± STD
LSNM-BB	$8.83{\cdot}10^{-3}$	0.8949	0.0040± 0.0025	$1.31{\cdot}10^{-3}$	0.8992	0.0010 ± 0.0000	$2.96{\cdot}10^{-3}$	0.9117	0.0011 ± 0.0005	$4.75{\cdot}10^{-3}$	0.9458	0.0090± 0.0033
GD-BB	$8.88{\cdot}10^{-1}$	0.0086	0.1565	$3.37{\cdot}10^{-3}$	0.8926	0.0009	$8.47{\cdot}10^{-2}$	0.9050	0.0030	$7.84{\cdot}10^{-15}$	0.9519	$1.49\cdot10^{-14}$
SGD	$9.95{\cdot}10^{-1}$	0.4926	0.4521± 0.0000	$5.99{\cdot}10^{-3}$	0.8890	0.0330 ± 0.0002	$5.89{\cdot}10^{-2}$	0.9050	0.0210 ± 0.0000	$9.72{\cdot}10^{-1}$	0.5013	1.8514± 0.0000
SARAH	$9.85{\cdot}10^{-1}$	0.4926	0.4478± 0.0000	$2.62{\cdot}10^{-3}$	0.8935	0.0144 ± 0.0000	$3.82{\cdot}10^{-2}$	0.9050	0.0136 ± 0.0000	$9.17{\cdot}10^{-1}$	0.5013	1.7468± 0.0000
SGD-MOM	$9.70{\cdot}10^{-1}$	0.4926	0.4409± 0.0000	$5.42{\cdot}10^{-3}$	0.8969	0.8942 ± 0.0000	$1.02{\cdot}10^{-2}$	0.9059	0.0037 ± 0.0000	$8.03{\cdot}10^{-1}$	0.5013	1.5293 ± 0.0000
ASM	$6.20{\cdot}10^{-3}$	0.8906	0.0028± 0.0003	$1.21{\cdot}10^{-3}$	0.8962	0.0009± 0.0005	$1.46{\cdot}10^{-2}$	0.9179	0.0050 ± 0.0001	$1.82{\cdot}10^{-2}$	0.9411	0.0349± 0.0016

Table 5: Comparison between different methods for objective function with NN loss terms; at the final iteration $\overline{K}$, the following performance measures are reported: $R_{\overline{K}}$ computed at the training set, the averaged accuracy *Acc* computed at the testing set and the averaged optimality gap at the training set ± the relative standard deviation (OG± STD).

loss terms respectively. In particular, we observe that, in Figure 7, the smallest value of the averaged optimality gap (computed at the training set) after 30 epochs is obtained by **SGD-MOM** method. This value is very similar to the one provided by **LSNM-BB**; the meaningful difference is that, for **SGD** and **SDG-MOM** methods, a best tuned set of hyperparameters was used, obtained with expensive trial and error procedures, while the behaviour of **LSNM-BB** Algorithm is not significantly dependent on its hyperparameters. We point out that 2-3 epochs are enough for the optimality gap to reach a value fairly close to the final value. With regard to the final value of the mini-batch size, we highlight that the **LSNM-BB** Algorithm provides an increase of the mini-batch

size which is not very relevant and lower than that required in **ASM**. About the accuracy (computed at the testing set), all the method are around the 90%. Figure 8, related to the objective function with non-convex loss terms, highlights that the more efficient method in term of optimality gap at the end of 30 epochs is **LSNM-BB** Algorithm. Also in this case the increase of the mini-batch size is limited and lower than that required by **ASM**. The accuracy is very similar for all the considered methods. Both Figures 7-8 shown that the optimality gap provided by **LSNM-BB** Algorithm, however, has a standard deviation that increases as the iterations increase; the method appears very efficient in the first 2-3 epochs. Tables 4-5 summarize the results obtained at the end of 30 epochs for all the test problems. These results confirm the previous remarks, highlighting that **LSNM-BB** Algorithm can be very competitive, despite the increase of the mini-batch size. The only exception is the case of the **GD-BB** method for the *RCV1* dataset; in the 30 iterations carried out, the backtracking procedure is never triggered and the ABB_{min} rule determines long steps which in the first approximately 20 iterations maintain the optimality gap comparable with the one of the other methods, while, in the last 10 iterations, this rule allows to reach a very low value of the optimality gap.
Now, in order to deepen the comparison of **LSNM-BB** Algorithm with the state of the art methods, we consider a similar method, as **SLiSeS** method in [11]; this method exhibits a particularly efficient numerical behavior in the early iterations. In the numerical experiment where the behaviour of **SLiSeS** method is compared with the one of **LSNM-BB** Algorithm over the first epoch, we used the Matlab code provided by the authors where **SLiSeS** method is equipped with the same setting specified in [11]; in particular, the stochastic gradient is related to a single term of the objective function, randomly drawn from the training set, and any inner cycle consists of 3 iterations. As in the numerical experiments in [11], we execute a single representative run of **SLiSeS** method and **LSNM-BB** Algorithm; to monitor the behaviour of both methods, the optimality gap evaluation at the end of any outer iteration is carried out.

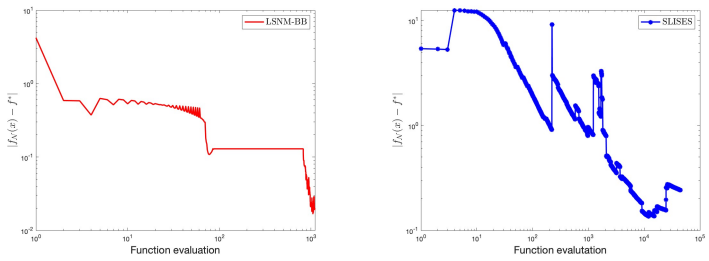

Figure 9: Dataset *w8a* and objective function with LR loss terms - Behaviour of the optimality gap computed at the training set with respect to the function evaluations in the first epoch for **LSNM-BB** on the left and for **SLiSeS** on the right. A logarithmic scale is used for both axes.

Table 6 shows the metrics for the two methods at the end of the first epoch; we observe that **LSNM-BB** Algorithm appears very efficient in term of optimality gap and accuracy for all test problems. Figures 9 and 10 show the comparison between the results obtained by the two methods for the *w8a* dataset and the objective function with convex and non-convex terms, respectively. We observe

		LSNM-BB		SLiSeS	
		LR	NN	LR	NN
MNIST	$R_{\overline{K}}$	$2.11 \cdot 10^{-3}$	$4.00 \cdot 10^{-2}$	*	$6.67 \cdot 10^{-1}$
	Acc	0.8791	0.8763	*	0.4927
	OG	0.0484	0.0182	*	0.3030
w8a	$R_{\overline{K}}$	$3.56 \cdot 10^{-3}$	$4.01 \cdot 10^{-3}$	$4.39 \cdot 10^{-2}$	$1.46 \cdot 10^{-1}$
	Acc	0.8935	0.8975	0.7741	0.7262
	OG	0.0196	0.0031	0.2419	0.1124
IJCNN	$R_{\overline{K}}$	$1.40 \cdot 10^{-2}$	$9.57 \cdot 10^{-3}$	$2.59 \cdot 10^{-1}$	$1.96 \cdot 10^{-1}$
	Acc	0.9099	0.9053	0.9037	0.9050
	OG	0.0121	0.0034	0.2242	0.0697
RCV1	$R_{\overline{K}}$	$2.03 \cdot 10^{-2}$	$1.56 \cdot 10^{-2}$	$7.13 \cdot 10^{-1}$	$4.80 \cdot 10^{-1}$
	Acc	0.9329	0.9284	0.5070	0.5013
	OG	0.0651	0.0297	2.2815	0.9142

Table 6: Comparison between **LSNM-BB** Algorithm and **SLiSeS** method for the two considered objective functions (with LR and NN loss terms); at the final iteration $\overline{K}$ of the first epoch, the following performance measures are reported: $R_{\overline{K}}$ computed at the training set, the accuracy *Acc* computed at the testing set and the optimality gap at the training set. The symbol * denotes a failure of the method.

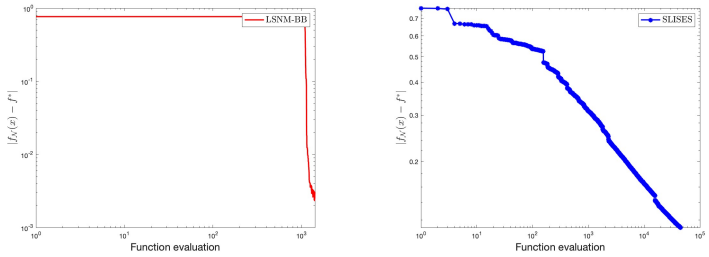

Figure 10: Dataset *w8a* and objective function with NN loss terms - Behaviour of the optimality gap computed at the training set with respect to the function evaluations in the first epoch for **LSNM-BB** on the left and for **SLiSeS** on the right. A logarithmic scale is used for both axes.

that, in the case of **SLiSeS** method, the optimality gap exhibits an almost linear decrease until the end of the epoch, while, in the case of **LSNM-BB**, a very rapid decrease occurs only at the end of the first epoch. The same results can be observed for the *RCV1* dataset and the objective function with convex and non-convex terms in Figures 12 -11 respectively. In summary, in the initial epochs the behavior of **LSNM-BB** Algorithm and **SLiSeS** method appears very similar: **LSNM-BB** Algorithm seems to determine a stepwise descent of the objective function while in the case of **SLiSeS** method the decrease seems more regular. However, at the end of the first epoch **LSNM-BB** Algorithm appears to perform better than **SLiSeS** method for all test problems considered. Finally, the last experiment is aimed to compare **LSNM-BB** Algorithm and **GD-BB** along 500 epochs. We remark that **LSNM-BB** can fall back to **GD-BB** method when the mini-batch is the whole dataset. In order to speed up this

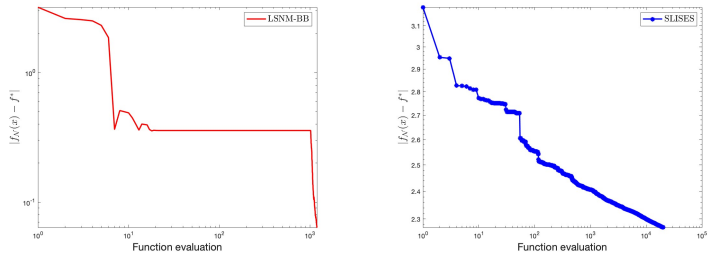

Figure 11: Dataset *RCV1* and objective function with LR loss terms - Behaviour of the optimality gap computed at the training set with respect to the function evaluations in the first epoch for **LSNM-BB** on the left and for **SLiSeS** on the right. A logarithmic scale is used for both axes.

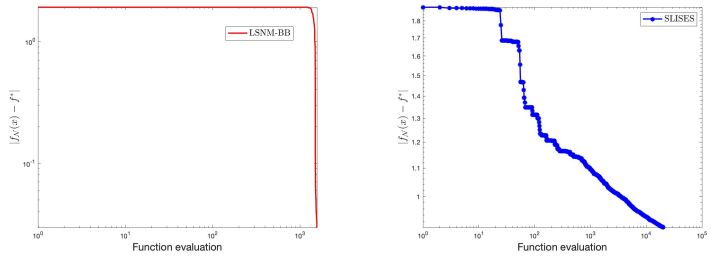

Figure 12: Dataset *RCV1* and objective function with NN loss terms - Behaviour of the optimality gap computed at the training set with respect to the function evaluations in the first epoch for **LSNM-BB** on the left and for **SLiSeS** on the right. A logarithmic scale is used for both axes.

process, at the step 22 of **LSNM-BB** Algorithm we set the rule of the increase of mini-batch size as $N_{k+1} = \min(N_k + 250, N)$.

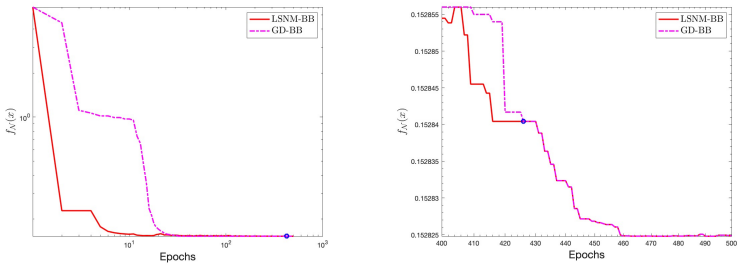

Figure 13: Dataset *w8a* and objective function with LR loss terms - On the left, behaviour of the objective function for **LSNM-BB** and **GD-BB** methods along 500 epochs; on the right a zoom along $400 - 500$ epochs is shown; at the epoch 426, **LSNM-BB** uses the whole dataset and the two methods exhibit the same behaviour.

Figures 13-14 show the behaviour of the objective function with LR and NN loss terms respectively for **LSNM-BB** and **GD-BB** methods along 500 epochs: we observe that after 426 epochs for the convex objective function and 312 epochs

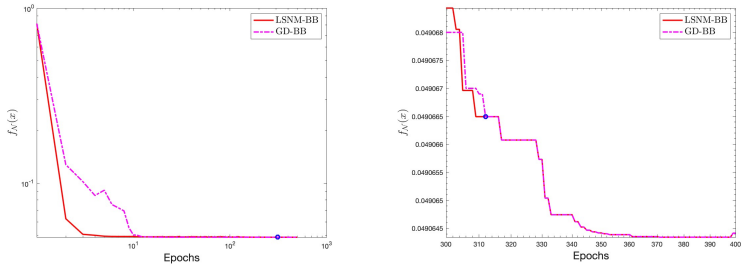

Figure 14: Dataset *w8a* and objective function with NN loss terms - On the left, behaviour of the objective function for **LSNM-BB** and **GD-BB** methods along 500 epochs; on the right a zoom along $300 - 400$ epochs is shown; at the epoch 312, **LSNM-BB** uses the whole dataset and the two methods exhibit the same behaviour.

for the non-convex one, **LSNM-BB** Algorithm falls into the method **GD-BB**, by using the whole training set in both the cases.
These results allow us to give a double interpretation of the performance of the proposed **LSNM-BB** Algorithm:

- for small-medium datasets with moderate memory demanding, the method can be used as a low-cost technique for finding a suitable starting point of any deterministic method;
- for very big datasets, the **LSNM-BB** Algorithm appears competitive with state of the art stochastic methods in terms of limited memory requirements and low computational costs; in addition, it does not require any hyperparameter tuning phase and appears robust with respect to the predefined rules that enable to control the maximum memory request in connection with the available computing resources.

5.4 A further numerical test simulating the infinite sum problem

In order to evaluate the behaviour of **LSNM-BB-G** Algorithm, we simulate the infinite sum problem (2). We consider the dataset *CIFAR10*, dowloadable from , where $d-1 = 3072$, the size of the training set is 50000 and the one of the testing set is 10000. We adapt *CIFAR10* for the binary case, where the two classes are the even and odd class positions. To mimic the behaviour of an incremental database, we follow the approach used in [20, 21, 22]. We subdivide the training and the testing sets in 30 blocks of 1666 and 333 elements respectively. Then, at the start of the code run, at 0 seconds, **LSNM-BB-G** Algorithm is executed by considering the first blocks of training and testing sets; every two seconds new blocks of the training and the testing sets are added to the previous ones, enlarging the two sets. So, at 0 seconds we have 1666 elements for the training and 333 elements for the testing, at 2 second we have 3332 elements for the training and 666 elements for the testing and so on. As error measure [22], we consider the quantity named *error rate*, given by $1-Acc$, where the accuracy is computed for the training and the testing sets before adding the new blocks at the end of two seconds. Every two seconds, the time count is stopped to evaluate

the metrics (error rate and objective function) and to increase the training and testing sets; then, it is restarted for the next two seconds. In **LSNM-BB-G** Algorithm, the rule for the increase of N_k and D_k is respectively $N_{k+1} = N_k + 1$ and $D_{k+1} = D_k + 1$ where $N_0 = 5$ and $D_0 = 1$.
In the following we describe the numerical results obtained by running **LSNM-BB-G** Algorithm on the described incremental training set. As in the previous experiments, we execute 10 runs of the code, changing the seed of the random number generator, and we report the averaged values of the measured quantities. In particular, the plots of the following averaged quantities are shown:

- objective function computed at the training set every two seconds before adding the new blocks;
- mini-batch size N_k with respect to the iterations;
- mini-batch size D_k with respect to the iterations;
- error rate for the training and the testing sets every two seconds before adding the new blocks.

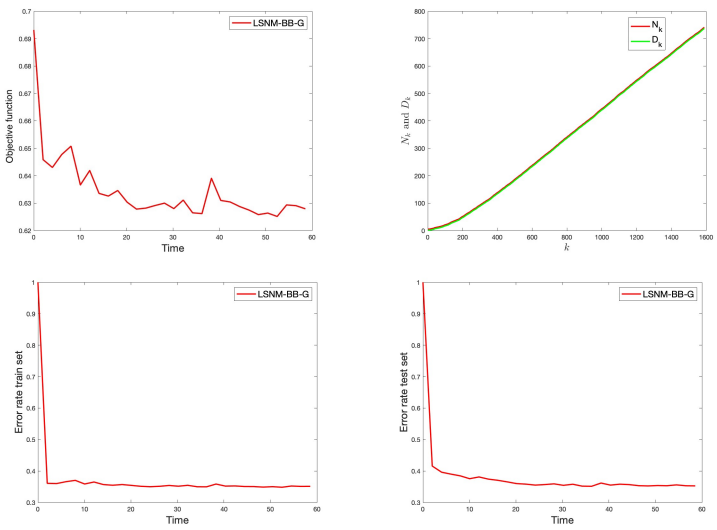

Figure 15: Behavior of **LSNM-BB-G** Algorithm. First row: on the left panel, objective function evaluated for the training set every two seconds; on the right panel, the increase of the mini-batch sizes N_k and D_k. Second row: on the left panels, error rate computed at the training set every two seconds; on the right panel, error rate computed at the testing set every two seconds.

In Figure (15) we observe that the objective function measured every 2 seconds on an increasing training set shows a decreasing trend. About the error rates measured on the testing and the training sets, we remark that they have a decreasing behaviour, very similar, achieving around the 40%. The increase of the mini-batch sizes N_k and D_k are very similar and limited under the value 1000. A difference with respect to **LSNM-BB** Algorithm is the increase of the size D_k; indeed, in **LSNM-BB** Algorithm, D_k is fixed to 1.

We remark that in the case of an incremental database the proposed algorithm appears stable; after the first learning phase based on the initial dataset, the procedure maintains or improves the accuracy, even after the successive increments of the database. These results are confirmed even if the increase of the training and the testing sets occurs with time intervals other than 2 seconds.

6 Conclusions

We developed a new method tailored for the numerical minimization of both finite sum (2) and objective functions in the form of mathematical expectation (1). This stochastic first-order method is based on a strategy to exploit the effectiveness of the well-known BB-like rules for the updating of the step length in the framework of the standard gradient descent methods; the idea is to group into cycles the iterations that consider the same mini-batch or SAA estimator of the objective function and to include in the current iteration an additional SD condition evaluated on a different mini-batch or SAA estimator; when this condition is not met, the trial iterate is rejected and the size of the new mini-batch or SAA estimator is increased; on the other hand, when all the iterations foreseen in a cycle are executed, a new mini-batch or SAA estimator with the same cardinality of the previous is chosen.
The details of the method are described in **LSNM-BB** and **LSNM-BB-G** Algorithms. Convergence results are discussed for the finite and infinite version for general and strongly convex objective function. Numerical experimentation for the binary classification of datasets well-known in the literature show very promising performance of the method, without the need to provide special values for the hyperparameters on which the method depends. Further future investigations can be planned to evaluate how to adapt the method to training neural networks and to handle incremental databases. Future experiments will allow evaluating the behaviour of the algorithm even when the oldest blocks are eliminated to limit memory requests.

Acknowledgments

We are grateful to the two anonymous referees whose constructive comments helped us to improve the paper. Furthermore, we thank the authors of [11] for making the SLiSeS Matlab code available to us.
NKJ and VR dedicate this work to Daniela Di Serafino, with whom they shared projects, ideas, research activities and an intense friendship.
The work of IT and VR work was supported by "Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM)" (Progetti 2023) and by European Union - NextGenerationEU through the Italian Ministry of University and Research as part of the PNRR – Mission 4 Component 2, Investment 1.3 (MUR Directorial Decree no. 341 of 03/15/2022), FAIR "Future" Partnership Artificial Intelligence Research", Proposal Code PE00000013 - CUP DJ33C22002830006). The work of NKJ was supported by the Science Fund of the Republic of Serbia, Grant no. 7359, Project LASCADO.

Availability of Data and Materials

The datasets analysed during the current study are available in links given in the paper.

Conflict of interest

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Liu, Y., Gao, Y. & Yin, W. An improved analysis of stochastic gradient descent with momentum. *Proceedings Of The 34th International Conference On Neural Information Processing Systems*. (2020)

[2] Krejić, N., Krklec Jerinkić, N. & Ostojić, T. Spectral projected subgradient method for nonsmooth convex optimization problems. *Numerical Algorithms*. **93**, 347-365 (2022)

[3] Ram, S., Nedic, A. & Veeravalli, V. Distributed stochastic subgradient projection algorithms for convex optimization. *J. Optim. Theory Appl.*. **147**, 516-545 (2011)

[4] Shapiro, A., Dentcheva, D. & Ruszczynski, A. Lectures on Stochastic Programming: Modeling and Theory. (SIAM,2009)

[5] Krejić, N., Krklec Jerinkić, N., Martínez, A. & Yousefi, M. A non-monotone trust-region method with noisy oracles and additional sampling. (2024)

[6] Frassoldati, G., Zanghirati, G. & Zanni, L. New Adaptive Stepsize Selections in Gradient Methods. *J. Ind. Manag. Optim.*. **4**, 299-312 (2008)

[7] Serafino, D., Ruggiero, V., Toraldo, G. & Zanni, L. On the steplength selection in gradient methods for unconstrained optimization. *Appl. Math. Comput.*. **318** pp. 176-195 (2018)

[8] Bottou, L., Curtis, F. & Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*. **60**, 223-311 (2018)

[9] Zhou, B., Gao, L. & Dai, Y. Gradient Methods with Adaptive Step-Sizes. *Comput. Optim. Appl.*. **35**, 69-86 (2006)

[10] Serafino, D., Krejić, N., Jerinkić, N. & Viola, M. LSOS: Line-search Second-Order Stochastic optimization methods for nonconvex finite sums. (2022)

[11] Bellavia, S., Krejić, N., Jerinkić, N. & Raydan, M. SLiSeS: Subsampled Line Search Spectral Gradient Method for Finite Sums. (2023)

[12] Grippo, L., Lampariello, F. & Lucidi, S. A nonmonotone line-search technique for Newton's method. *SIAM Journal On Numerical Analysis*. **23**, 707-716 (1986)

[13] Ferreira, O., Lemes, M. & Prudente, L. On the inexact scaled gradient projection method. *Comput. Optim. Appl.*. **81** pp. 91-125 (2022)

[14] Barzilai, J. & J.M.Borwein Two point step size gradient methods. *IMA J.Numer.Anal.*. **8** pp. 141-148 (1988)

[15] Klenke, A. Probability Theory: A Comprehensive Course. (Springer,2007)

[16] Birgin, E., Martìnez, J. & Raydan, M. Nonmonotone Spectral Projected Gradient Methods on Convex Sets. *SIAM Journal On Optimization*. **10**, 1196-1211 (2000)

[17] Raydan, M. The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem. *SIAM Journal On Optimization*. **7**, 26-33 (1997)

[18] Dai, Y. On the nonmonotone line search. *J. Optim. Theory Appl.*. **112** pp. 315-330 (2002)

[19] Dai, Y. & Liao, L. R-linear convergence of the Barzilai and Borwein gradient method. *IMA Journal Of Numerical Analysis*. **22**, 1-10 (2002)

[20] Tavakolizadeh, F., Soto, J., Gyulai, D. & Beecks, C. Industry 4.0: Mining Physical Defects in Production of Surface-Mount Devices. (2017), https://api.semanticscholar.org/CorpusID:117308853

[21] J. A. Carvajal Soto, F. & Gyulai, D. An online machine learning framework for early detection of product failures in an Industry 4.0 context. *International Journal Of Computer Integrated Manufacturing*. **32**, 452-465 (2019),

[22] Agarwal, S., Vijaya Saradhi, V. & Karnick, H. Kernel-based online machine learning and support vector reduction. *Neurocomputing*. **71**, 1230-1237 (2008), https://www.sciencedirect.com/science/article/pii/S0925231208000581, Progress in Modeling, Theory, and Application of Computational Intelligenc

[23] Krejić, N. & Krklec Jerinkić, N. Non-monotone line search methods with variable sample size. *Numerical Algorithms*. **68**, 711-739 (2015)

[24] Nguyen, L., Liu, J., Scheinberg, K. & Takáč, M. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. (2017)

[25] Bollapragada, R., Byrd, R. & Nocedal, J. Adaptive sampling strategies for stochastic optimization. *SIAM Journal On Optimization*. **28**, 3312-3343 (2018)