A STOCHASTIC FIRST-ORDER TRUST-REGION METHOD WITH INEXACT RESTORATION FOR FINITE-SUM MINIMIZATION[‡]

STEFANIA BELLAVIA[*], NATAŠA KREJIĆ[†], BENEDETTA MORINI[*], SIMONE REBEGOLDI[*]

Abstract. We propose a stochastic first-order trust-region method with inexact function and gradient evaluations for solving finite-sum minimization problems. At each iteration, the function and the gradient are approximated by sampling. The sample size in gradient approximations is smaller than the sample size in function approximations and the latter is determined using a deterministic rule inspired by the inexact restoration method, which allows the decrease of the sample size at some iterations. The trust-region step is then either accepted or rejected using a suitable merit function, which combines the function estimate with a measure of accuracy in the evaluation. We show that the proposed method eventually reaches full precision in evaluating the objective function and we provide a worst-case complexity result on the number of iterations required to achieve full precision. We validate the proposed algorithm on nonconvex binary classification problems showing good performance in terms of cost and accuracy and the important feature that a burdensome tuning of the parameters involved is not required.

Keywords: finite-sum minimization, inexact restoration, trust-region methods, subsampling, worst-case evaluation complexity.

AMS subject classifications. 65K05, 90C26, 68T05.

1. Introduction. In this paper we consider the following finite-sum minimization problem

$$\min_{x\in\mathbb{R}^n} f_N(x) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(x), \tag{1.1}$$

where N is very large and finite and $\phi_i : \mathbb{R}^n \to \mathbb{R}$, $1 \le i \le N$. We will assume throughout the paper that all the functions ϕ_i are continuously differentiable. A number of important problems can be stated in this form, e.g., classification problems in machine learning, data fitting problems, sample average approximations of an objective function given in the form of mathematical expectation. The need for efficient methods for solving (1.1) resulted in a large body of literature in the recent years and a number of methods have been proposed and analyzed, see e.g., the reviews [2,10,17].

A number of methods employ subsampled approximations of the objective function and its derivatives, with the aim of reducing the computational cost. Focusing on first-order methods, the stochastic gradient [26] and more contemporary variants like SVRG [19,20], SAG [27], ADAM [21] and SARAH [24] are widely used for their simplicity and low cost per-iteration. They do not call for function evaluations but require

[*]Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale G.B. Morgagni 40, 50134 Firenze, Italia. Members of the INdAM Research Group GNCS. Emails: stefania.bellavia@unifi.it, benedetta.morini@unifi.it, simone.rebegoldi@unifi.it

[†]Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia, Email: natasak@uns.ac.rs.

[‡]The research that led to the present paper was partially supported by a grant of the group GNCS of INdAM and partially developed within the Mobility Project: "Second order methods for optimization problems in Machine Learning" (ID: RS19MO05) executive programme of Scientic and Technological cooperation between the Italian Republic and the Republic of Serbia 2019-2022. The work of the second author was supported by Serbian Ministry of Education, Science and Technological Development, grant no. 451-03-9/2021-14/200125 and her visit to the Department of Industrial Engineering at the University of Florence (Italy), has been supported by the 2019 University of Florence International plan.

tuning the learning rate and further possible hyper-parameters such as the mini-batch size. Since the tuning effort may be very computationally demanding [15], more sophisticated approaches use linesearch or trust-region strategies to adaptively choose the learning rate and to avoid tuning efforts, see [2, 4, 5, 9, 14, 15, 25]. In this context, function and gradient approximations have to satisfy sufficient accuracy requirements with some probability. This, in turn, in case of approximations via sampling, requires adaptive choices of the sample sizes used.

In a further stream of works, problem (1.1) is reformulated as a constrained optimization problem and the sample size is computed deterministically using the Inexact Restoration (IR) approach. The IR approach has been successfully combined with either the linesearch strategy [22] or the trust-region strategy [3, 7, 8]; in these papers, function and gradient estimates are built with gradually increasing accuracy and averaging on the same sample.

In the following, we propose a novel trust-region method with random models based on the IR methodology. Differently from [3, 7, 8, 22], the accuracy in random function evaluations is gradually increased throughout the iterations while subsampled gradient approximations are used at each iteration. Before reaching full accuracy in function evaluations, the IR strategy gives rise to a deterministic rule for determining the sample size in function approximations, which is allowed to decrease at some iterations. Gradient approximations are computed using samples that are smaller than the ones used for function evaluations. We show that eventually full accuracy in function evaluations is reached and we provide an upper bound on the number of iterations in which the method uses function estimates averaged on a subsample of $\{1, \dots, N\}$. Once full accuracy in function evaluations is reached, our method falls into the STochastic Optimization framework with Random Models denoted as STORM and given in [1, 14]. More precisely, it reduces to a particular instance of STORM where random gradients and exact function values are used.

Notably, the IR approach smoothly drives the transition to STORM and numerical experience on nonconvex optimization problems arising from binary classification show that our procedure provides accurate classifications even if the process is terminated before reaching full accuracy in function evaluations, i.e. before reducing to STORM.

The paper is organized as follows. In Section 2 we give an overview of random models employed in the trust-region framework and discuss some sample size choices. The new algorithm is proposed in Section 3 and studied theoretically also with respect to its complexity analysis. Extensive numerical results are presented in Section 4.

2. Trust-region method and random models. Stochastic variants of standard trust-region methods based on the use of random models have been presented in [1,9,13,14,28]. They consist in the adaptation of the trust-region framework to the case where random estimates of the derivatives are introduced and function values are either computed exactly [1] or replaced by stochastic estimates [9, 13, 14, 28].

Here we focus on STORM (STochastic Optimization with Random Models) studied in [1,9,14], since eventually our method falls in such framework. The computation and acceptance of the iterates parallel the standard trust-region mechanism and the success of the procedure relies on function values and models being sufficiently accurate with fixed and large enough probability.

Let $\|\cdot\|$ denote the 2-norm throughout the paper. At iteration k of the first-order version of STORM, given the trust-region radius Δ_k and a random approximation g_k

of $\nabla f_N(x_k)$, consider the model

$$\varsigma_k(x_k + s) = f_N(x_k) + g_k^T s$$

for f_N on $B(x_k, \Delta_k) = \{x \in \mathbb{R}^n : \|x - x_k\| \leq \Delta_k\}$ and the trust-region problem $\min_{\|s\| \leq \Delta_k} \varsigma_k(x_k + s)$. Then, the trust region step takes the form $s_k = -\Delta_k g_k / \|g_k\|$ and two estimates $f^{k,0}$ and $f^{k,s}$ of f_N at x_k and $x_k + s_k$, respectively, are employed to either accept or reject the trial point $x_k + s_k$. The classical ratio between the actual and predicted reduction is replaced by

$$\rho_k = \frac{f^{k,0} - f^{k,s}}{\varsigma_k(x_k) - \varsigma_k(x_k + s_k)} \tag{2.1}$$

and a successful iteration is declared when $\rho_k \geq \eta_1$ and $\|g_k\| \geq \eta_2 \Delta_k$ for some constants $\eta_1 \in (0, 1)$ and η_2 positive and possibly large. Note that the computation of both the step s_k and the denominator in (2.1) are independent of $f_N(x_k)$. Furthermore, note that a successful iteration may not yield an actual reduction in f_N because the quantities involved in ρ_k are random approximations to the true value of the objective function. At this regard, the condition $\|g_k\| \geq \eta_2 \Delta_k$ serves as a guess of the true function decrease since Δ_k controls the accuracy of function and gradient estimates as follows.

The models used are required to be sufficiently accurate with some probability. Specifically, the model ς_k is supposed to be, p_M-probabilistically, a κ-fully linear model of f_N on the ball $B(x_k, \Delta_k)$, i.e., the requirement

$$|f_N(y) - \varsigma_k(y)| \leq \kappa \Delta_k^2, \quad \|\nabla f_N(y) - g_k\| \leq \kappa \Delta_k, \quad y \in B(x_k, \Delta_k) \tag{2.2}$$

has to be fulfilled at least with probability $p_M \in (0, 1)$. Moreover, the estimates $f^{k,0}$ and $f^{k,s}$ are supposed to be p_f-probabilistically ϵ_F-accurate estimates of $f_N(x_k)$ and $f_N(x_k + s_k)$, i.e., the requirement

$$|f^{k,0} - f_N(x_k)| \leq \epsilon_F \Delta_k^2, \quad |f^{k,s} - f_N(x_k + s_k)| \leq \epsilon_F \Delta_k^2, \tag{2.3}$$

has to be fulfilled at least with probability $p_f \in (0, 1)$. Clearly, if f is computed exactly then condition (2.3) is trivially satisfied.

Convergence analysis shows that for $p_M > \frac{1}{2}$, $p_f > \frac{1}{2}$ and p_M, p_f sufficiently large, it holds $\lim_{k \to \infty} \Delta_k = 0$ almost surely, see [14, Corollary 4.12]. Moreover, if f_N is bounded from below and ∇f_N is Lipschitz continuous, then the generated random sequence $\{x_k\}$ is such that $\lim_{k \to \infty} \|\nabla f_N(x_k)\| = 0$ almost surely, see [14, Theorem 4.18]. Interestingly, the accuracy in (2.2) and (2.3) increases as the trust region radius gets smaller but the probabilities p_M and p_f are fixed.

For problem (1.1) it is straightforward to build approximations of f_N and ∇f_N by sample average approximations

$$f_M(x) = \frac{1}{M} \sum_{i \in I_M} \phi_i(x), \qquad \nabla f_S(x) = \frac{1}{S} \sum_{i \in I_S} \nabla \phi_i(x), \tag{2.4}$$

where I_M and I_S are subsets of $\{1, \ldots, N\}$ of cardinality $|I_M| = M$ and $|I_S| = S$, respectively.

The issue of choosing the sample size such that (2.2) and (2.3) hold in probability is delicate. Dynamic strategies for choosing the sample size have been proposed, see e.g., [4, 12, 14, 25]. Considering, for sake of brevity, the estimate $f^{k,0} \stackrel{\text{def}}{=}$

$\frac{1}{M}\sum_{i\in I_M}\phi_i(x_k)$ for $f_N(x_k)$, results in [5,14,25] indicate that $f^{k,0}$ is a p_f-probabilistically accurate estimate of $f_N(x_k)$ if

$$|M| \geq \mathcal{O}\left(\frac{V_f}{\epsilon_F \Delta_k^2}\right) \quad \text{and} \quad |M| \leq N,$$

where V_f is such that $\mathbb{E}[|\phi_i(x) - f_N(x)|^2] \leq V_f$, $i = 1, \ldots, N$, $\mathbb{E}$ indicates the expected value of a random variable and $\mathcal{O}$ hides the log factor of $1/(1-p_f)$.

Analogous results hold for the sample average approximation $f^{k,s}$ of $f_N(x_k + s_k)$ and the sample average approximation g_k of $\nabla f_N(x_k)$ [14, Section 5], [15]. In particular, note that if ∇f_N is Lipschitz continuous and $\|\nabla f_N(x_k) - g_k\| \leq \kappa\Delta_k$ then (2.2) holds.

In principle, conditions (2.2), (2.3) and $\lim_{k\to\infty}\Delta_k = 0$ imply that g_k, $f^{k,0}$ and $f^{k,s}$ will be computed at full precision for k sufficiently large. On the other hand, in applications such as machine learning, reaching full precision is unlikely since N is very large and termination is based on the maximum allowed computational effort or on the validation error.

Applying the above estimate for the sample size is not trivial since in general the upper bound V_f on the variance is not available; however it can be replaced with variance estimates [12] obtained during the computation of the subsampled function values, or with estimated upper bounds on $|f_N(x_k)|$ and $\|\nabla f_N(x_k)\|$ [4, 29]. Alternative proposals consist in increasing the sample size by a prefixed factor, or geometrically with the iteration index k by a rule of the form $\lceil a^k \rceil$ for some $a > 1$, [11, 12]. As for g_k, the sample size can be fixed using a specific inner product test that ensures that it is a descent direction with high probability [11]. Finally, we observe that experiments with constant sample sizes are reported in [12, 14] but Byrd et al., [12] observed that most of the improvement in the objective function occurred with a dynamic choice of the sample size.

The main contribution of our work consists in the definition of a novel trust-region method with random models where the sample size of function approximations is computed through a deterministic rule inspired by the inexact restoration method [23], instead of the probabilistic accuracy requirement (2.3); gradient approximations are required to satisfy (2.2) only after full precision in function evaluations is achieved. We introduce the novel trust-region method in the next section.

3. The Algorithm. In this section we introduce our trust-region method with random models. The sample size used for the estimate of $f_N(x_k)$ changes dynamically along the iterations and is adjusted by a deterministic rule inspired by the inexact restoration (IR) method [23].

In order to employ the IR framework, we make a simple transformation of (1.1) into a constrained problem and then apply the IR strategy combined with a trust-region method. Letting I_M be an arbitrary nonempty subset of $\{1, \ldots, N\}$ of cardinality $|I_M|$ equal to M, we reformulate problem (1.1) as

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f_M(x) = \frac{1}{M}\sum_{i\in I_M}\phi_i(x), \\ &\text{s.t. } M = N. \end{aligned} \tag{3.1}$$

Given the reformulated problem, we measure the level of infeasibility with respect to the constraint $M = N$ by the following function h.

ASSUMPTION 3.1. *Let $h : \{1, 2, \ldots, N\} \to \mathbb{R}$ be a monotonically decreasing function such that $h(1) > 0$, $h(N) = 0$.*

This assumption implies that there exist some positive $\underline{h}$ and $\overline{h}$ such that

$$\underline{h} \leq h(M) \quad \text{if} \quad 0 < M < N, \qquad \text{and} \qquad h(M) \leq \overline{h} \quad \text{if} \quad 0 < M \leq N, \tag{3.2}$$

for $M \in \mathbb{N}$. One possible choice is $h(M) = (N - M)/N,\ 0 < M \leq N$.

The IR methods improve feasibility and optimality in modular way using a merit function to balance the progress. Since the reductions in the objective function and infeasibility might be achieved to a different degree, the IR method employs the merit function

$$\Psi(x, M, \theta) = \theta f_M(x) + (1 - \theta)h(M), \tag{3.3}$$

with $\theta \in (0, 1)$.

Let us now describe the main ingredients of the proposed algorithm, referred to as SIRTR (Stochastic Inexact Restoration Trust Region). SIRTR is a trust-region method that employs first-order random models. In particular, at a generic iteration k, given a random direction g_k, we fix a *trial* sample size N_{k+1}^t and build a linear model $m_k(p)$ around x_k of the form

$$m_k(p) = f_{N_{k+1}^t}(x_k) + g_k^T p. \tag{3.4}$$

Then, we consider the trust-region problem

$$\min_{\|p\| \leq \Delta_k} m_k(p) \tag{3.5}$$

whose solution is

$$p_k = -\Delta_k \frac{g_k}{\|g_k\|}. \tag{3.6}$$

As in standard trust-region methods, we distinguish between successful and unsuccessful iterations. However, we do not employ here the classical acceptance condition, but a more elaborate one that involves the merit function (3.3). The proposed method is sketched in Algorithm 3.1.

We first describe the scheduling procedure for generating the trial sample size N_{k+1}^t. At a generic iteration k, we have at hand the outcome of the previous iteration: the iterate x_k, the sample sizes N_k and $\widetilde{N}_k$, the penalty parameter θ_k and the flags $\mathcal{F}_{k-1}$ and `iflag`. If `iflag=succ` the previous iteration was successful, i.e. $x_k = x_{k-1} + p_{k-1}$, whereas `iflag=unsucc` indicates that it was unsuccessful, i.e. $x_k = x_{k-1}$. At Step 1 of SIRTR, if `iflag=succ`, we set $\mathcal{F}_k = 0$ or $\mathcal{F}_k = 1$ accordingly to whether $N_k < N$ or $N_k = N$, and compute a new sample size $\widetilde{N}_{k+1} \leq N$ such that the infeasibility measure h is sufficiently decreased as stated in (3.7); note that, if $N_k = N$, then (3.7) automatically implies $\widetilde{N}_{k+1} = N$. If `iflag=unsucc`, we set $\widetilde{N}_{k+1} = \widetilde{N}_k$ and $\mathcal{F}_k = \mathcal{F}_{k-1}$. In principle we could set the trial sample size to $\widetilde{N}_{k+1}$; however, since we aim at reducing the computational effort, at Step 2 we choose a trial sample size N_{k+1}^t satisfying $N_{k+1}^t \leq \widetilde{N}_{k+1}$ and condition (3.8). Thus the distance between the sample size used to approximate the objective function and $\widetilde{N}_{k+1}$ is related to the trust-region size Δ_k.

Algorithm 3.1: The Stochastic IRTR algorithm

Given $x_0 \in \mathbb{R}^n$, N_0 integer in $(0, N]$, $\theta_0 \in (0,1)$, $0 < \Delta_0 < \Delta_{\max}$,
$\gamma > 1$, $r, \eta, \in (0,1)$, μ, $\eta_2 > 0$, $\underline{h}$ given in (3.2).

0. Set $k = 0$, `iflag=succ`.
1. If `iflag=succ`
   If $N_k < N$ set $\mathcal{F}_k = 0$, else set $\mathcal{F}_k = 1$.
   Find $\widetilde{N}_{k+1}$ such that $N_k \leq \widetilde{N}_{k+1} \leq N$ and

$$h(\widetilde{N}_{k+1}) \leq r h(N_k), \tag{3.7}$$

   Else set $\widetilde{N}_{k+1} = \widetilde{N}_k$, $\mathcal{F}_k = \mathcal{F}_{k-1}$.
2. Find N^t_{k+1} such that $N^t_{k+1} \leq \widetilde{N}_{k+1}$ and

$$h(N^t_{k+1}) - h(\widetilde{N}_{k+1}) \leq \mu \Delta_k^2. \tag{3.8}$$

3. Choose the search direction $g_k \in \mathbb{R}^n$, build $m_k(p) = f_{N^t_{k+1}}(x_k) + g_k^T p$, and set

$$p_k = -\Delta_k \frac{g_k}{\|g_k\|}.$$

4. If $N_k = N$, $N^t_{k+1} < N$ and

$$f_N(x_k) - m_k(p_k) < \Delta_k \|g_k\|, \tag{3.9}$$

   take $\Delta_k = \Delta_k/\gamma$ and go to Step 2.
5. Compute the penalty parameter θ_{k+1}

$$\theta_{k+1} = \begin{cases} \theta_k & \text{if } \ \mathrm{Pred}_k(\theta_k) \geq \eta(h(N_k) - h(\widetilde{N}_{k+1})) \\ \dfrac{(1-\eta)(h(N_k) - h(\widetilde{N}_{k+1}))}{m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1})} & \text{otherwise.} \end{cases} \tag{3.10}$$

6. If $\mathrm{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta \mathrm{Pred}_k(\theta_{k+1})$ and $(\|g_k\| - \eta_2 \Delta_k)\mathcal{F}_k \geq 0$
   (*successful iteration*) define

$$x_{k+1} = x_k + p_k$$

$$\Delta_{k+1} = \begin{cases} \min\{\gamma \Delta_k, \Delta_{\max}\}, & \text{if } h(N_k) - h(\widetilde{N}_{k+1}) = 0 \text{ and } \Delta_k^2 \geq \underline{h}/\mu \\ \Delta_k, & \text{otherwise} \end{cases} \tag{3.11}$$

   set $N_{k+1} = N^t_{k+1}$, $k = k+1$, `iflag=succ` and go to Step 1.
   Else (*unsuccessful iteration*) set $x_{k+1} = x_k$, $N_{k+1} = N_k$,
   $\Delta_{k+1} = \Delta_k/\gamma$, $k = k+1$, `iflag=unsucc` and go to Step 1.

At Step 3 we choose the search direction g_k and compute the solution p_k of the

trust-region subproblem (3.5). As for the model $m_k(p)$ in (3.4), we compute

$$f_{N^t_{k+1}}(x_k) = \frac{1}{N^t_{k+1}} \sum_{i \in I_{N^t_{k+1}}} \phi_i(x_k). \tag{3.12}$$

The set of samples $I_{N^t_{k+1}} \subseteq \{1, \ldots, N\}$ of cardinality N^t_{k+1} is meant to be chosen randomly in practice, even though no particular computational rule for $I_{N^t_{k+1}}$ (stochastic or deterministic) will be assumed in the convergence analysis.

Step 4 is designed to take care of the special case when we are at the full sample size $N_k = \widetilde{N}_{k+1} = N$, but the trial sample size N^t_{k+1} from Step 2 is smaller than N, i.e., we are using a cheaper approximate objective function than f_N. Such decrease in precision is meaningful only if the linear model $m_k(p)$ given in (3.4) is a good approximation of f_N. Therefore we check the quality of the linear model through (3.9) and either continue with $N^t_{k+1} < N$ or we decrease the trust region size and return to Step 2, where (3.8) gives rise to a new N^t_{k+1}. Notice that the loop in Steps 2-4 has finite termination. Indeed, due to (3.8), it follows that $N^t_{k+1} = \widetilde{N}_{k+1}$ for Δ_k sufficiently small, Step 4 is not performed and the method goes through Step 5. At Step 5 we compute the new value of the penalty term θ_{k+1}. The computation relies on the definition of the predicted reduction defined as

$$\mathrm{Pred}_k(\theta) = \theta(f_{N_k}(x_k) - m_k(p_k)) + (1-\theta)(h(N_k) - h(\widetilde{N}_{k+1})), \tag{3.13}$$

where $\theta \in (0,1)$. This definition is a convex combination of the usual part for trust-region methods, $f_{N_k}(x_k) - m_k(p_k)$, and the predicted infeasibility $h(N_k) - h(\widetilde{N}_{k+1})$ obtained in Step 1. The new parameter θ_{k+1} is computed in Step 5 so that the following condition on the predicted reduction is satisfied

$$\mathrm{Pred}_k(\theta) \geq \eta(h(N_k) - h(\widetilde{N}_{k+1})). \tag{3.14}$$

If (3.14) is satisfied for $\theta = \theta_k$ then $\theta_{k+1} = \theta_k$, otherwise θ_{k+1} is computed as the largest value for which the above inequality holds (see forthcoming Lemma 3.2).

In order to decide if the iteration is successful or not, we first define the actual reduction as follows. Given a point $\hat{x}$ and $\theta \in (0,1)$, the actual reduction of Ψ at the point $\hat{x}$ has the form

$$\begin{aligned} \mathrm{Ared}_k(\hat{x}, \theta) &= \Psi(x_k, N_k, \theta) - \Psi(\hat{x}, N^t_{k+1}, \theta) \\ &= \theta(f_{N_k}(x_k) - f_{N^t_{k+1}}(\hat{x})) + (1-\theta)(h(N_k) - h(N^t_{k+1})). \end{aligned} \tag{3.15}$$

At Step 6 we declare the iteration successful whenever the following two conditions are both satisfied

$$\mathrm{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta \mathrm{Pred}_k(\theta_{k+1}) \tag{3.16}$$

$$(\|g_k\| - \eta_2 \Delta_k)\mathcal{F}_k \geq 0, \tag{3.17}$$

otherwise the iteration is declared unsuccessful. Again, $f_{N^t_{k+1}}(x_k + p_k)$ in $\mathrm{Ared}_k(x_k + p_k, \theta_{k+1})$ is computed using (2.4) and the set of samples $I_{N^t_{k+1}}$. On one hand, condition (3.16) mimics the classical acceptance criterion of standard trust-region methods, with the substantial difference that $\mathrm{Ared}_k(x_k + p_k, \theta_{k+1})$ involves the merit function (3.3) in place of f_N and $\mathrm{Pred}_k(\theta_{k+1})$ involves the infeasibility measure $h(N_k) - h(\widetilde{N}_{k+1})$

in addition to the classical predicted reduction. On the other hand, condition (3.17) is actually imposed only if $\mathcal{F}_k = 1$, i.e. $N_k = N$; in this case, it can be written as $\|g_k\| \geq \eta_2 \Delta_k$. Thanks to this condition, Δ_k is driven to zero as $\|g_k\|$ tends to zero, $N_{k+1}^t = N$ for sufficiently large k and our procedure reduces to STORM (see Section 3.1). If the iteration is successful, we accept the step and the trial sample size, set `iflag=succ` and adjust the trust-region radius through (3.11), which allows to increase the trust region size only if $N_k = \widetilde{N}_{k+1} = N$ and $\Delta_k \geq \underline{h}/\mu$. The upper bound $\Delta_{\max}$ on the trust region size is imposed in (3.11). In case of unsuccessful iterations, we reject both the step and the trial sample size, set `iflag=unsucc` and decrease the trust region size.

We conclude the description of Algorithm 3.1 showing that condition (3.7) holds for all iterations, even when it is not explicitly enforced at Step 1.

LEMMA 3.1. *Let Assumption 3.1 holds. The sample sizes $\widetilde{N}_{k+1} \leq N$ and $N_k \leq N$ generated by Algorithm 3.1 satisfy*

$$h(\widetilde{N}_{k+1}) \leq r h(N_k), \quad \forall k \geq 0. \tag{3.18}$$

Proof. We observe that, by Assumption 3.1, (3.18) trivially holds whenever $N_k = \widetilde{N}_{k+1} = N$.

Otherwise, we proceed by induction. Indeed, the thesis trivially holds for $k = 0$, as we set `iflag=succ` at the first iteration and thus enforce (3.18) at Step 1. Now consider a generic iteration $\bar{k}$ and suppose that (3.18) holds for $\bar{k} - 1$. If $\bar{k} - 1$ is successful, then condition (3.18) is enforced for iteration $\bar{k}$ at Step 1. If $\bar{k} - 1$ is unsuccessful, then we reject the trial sample size $N_{\bar{k}}^t$, and set $N_{\bar{k}} = N_{\bar{k}-1}$ at Step 6 of iteration $\bar{k} - 1$ and $\widetilde{N}_{\bar{k}+1} = \widetilde{N}_{\bar{k}}$ at Step 1 of iteration $\bar{k}$. Since (3.18) holds by induction at iteration $\bar{k} - 1$, we have $h(\widetilde{N}_{\bar{k}}) \leq r h(N_{\bar{k}-1})$, which can be rewritten as $h(\widetilde{N}_{\bar{k}+1}) \leq r h(N_{\bar{k}})$ due to the assignments at Step 6 and Step 1. Then condition (3.18) holds also at iteration $\bar{k}$. □

3.1. On the behaviour of $\{N_k\}$ and $\{\Delta_k\}$. In this section, we analyze the properties of Algorithm 3.1. In particular, we prove that the sequence $\{\theta_k\}$ is nonincreasing and uniformly bounded from below, the trust region radius Δ_k tends to 0 as $k \to \infty$ and $N_k = N$ for all k sufficiently large. In the analysis that follows we will consider two options for $\hat{x}$ in (3.15), $\hat{x} = x_k + p_k$ for successful iterations and $\hat{x} = x_k$ for unsuccessful iterations.

ASSUMPTION 3.2. *There exist $\Omega \subset \mathbb{R}^n$ and f_{low}, f_{up} such that*

$$f_{low} < f_M(x) < f_{up}, \quad \forall\ 1 \leq M \leq N,\ x \in \Omega,$$

and all iterates generated by Algorithm 3.1 belong to Ω.

In the following, we let

$$\kappa_\phi = \max\{|f_{low}|, |f_{up}|\}. \tag{3.19}$$

In the context of machine learning, the above assumption is verified in several cases, e.g., the mean-squares loss function coupled with either the sigmoid, the softmax or the hyperbolic tangent activation function; the mean-squares loss function coupled with ReLU or ELU activation functions and proper bounds on the unknowns; the logistic loss function coupled with proper bounds on the unknowns [18].

Our first result characterizes the sequence $\{\theta_k\}$ of the penalty parameters; the proof follows closely [3, Lemma 2.2].

LEMMA 3.2. *Let Assumptions 3.1 and 3.2 hold. Then the sequence* $\{\theta_k\}$ *is positive, nonincreasing and bounded from below,* $\theta_{k+1} \geq \underline{\theta} > 0$ *with* $\underline{\theta}$ *independent of* k *and* (3.14) *holds with* $\theta = \theta_{k+1}$.

Proof. We note that $\theta_0 > 0$ and proceed by induction assuming that θ_k is positive and that for all iterations k, due to Step 1, we have that $N_k \leq \widetilde{N}_{k+1}$ and $N_k = \widetilde{N}_{k+1}$ if and only if $N_k = N$. First consider the case where $N_k = \widetilde{N}_{k+1}$ (or equivalently $N_k = \widetilde{N}_{k+1} = N$); then it holds $h(N_k) - h(\widetilde{N}_{k+1}) = 0$. If $N_{k+1}^t < N$, due to Step 4, we have $\mathrm{Pred}_k(\theta) = \theta(f_{N_k}(x_k) - m_k(p_k)) > 0$ for any positive θ. Otherwise, if $N_{k+1}^t = N$, then $\mathrm{Pred}_k(\theta) = \theta\Delta_k\|g_k\| > 0$ for any positive θ. Thus, in both cases (3.10) implies $\theta_{k+1} = \theta_k$.

Let us now consider the case $N_k < \widetilde{N}_{k+1}$. If inequality $\mathrm{Pred}_k(\theta_k) \geq \eta(h(N_k) - h(\widetilde{N}_{k+1}))$ holds then (3.10) gives $\theta_{k+1} = \theta_k$. Otherwise, we have

$$\theta_k \left(f_{N_k}(x_k) - m_k(p_k) - (h(N_k) - h(\widetilde{N}_{k+1}))\right) < (\eta - 1)\left(h(N_k) - h(\widetilde{N}_{k+1})\right),$$

and since the right hand-side is negative by assumption, it follows

$$f_{N_k}(x_k) - m_k(p_k) - (h(N_k) - h(\widetilde{N}_{k+1})) < 0.$$

Consequently, $\mathrm{Pred}_k(\theta) \geq \eta(h(N_k) - h(\widetilde{N}_{k+1}))$ is satisfied if

$$\theta(f_{N_k}(x_k) - m_k(p_k) - (h(N_k) - h(\widetilde{N}_{k+1}))) \geq (\eta - 1)(h(N_k) - h(\widetilde{N}_{k+1})),$$

i.e., if

$$\theta \leq \theta_{k+1} \stackrel{\mathrm{def}}{=} \frac{(1-\eta)(h(N_k) - h(\widetilde{N}_{k+1}))}{m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1})}.$$

Hence θ_{k+1} is the largest value satisfying (3.14) and $\theta_{k+1} < \theta_k$.

Let us now prove that $\theta_{k+1} \geq \underline{\theta}$. Note that by (3.18) and (3.2)

$$h(N_k) - h(\widetilde{N}_{k+1}) \geq (1-r)h(N_k) \geq (1-r)\underline{h}. \tag{3.20}$$

Thus, using (3.19)

$$\begin{aligned} m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1}) &\leq m_k(p_k) - f_{N_k}(x_k) + h(N_k) \\ &\leq f_{N_{k+1}^t}(x_k) - \Delta_k\|g_k\| - f_{N_k}(x_k) + \overline{h} \\ &\leq |f_{N_{k+1}^t}(x_k) - f_{N_k}(x_k)| + \overline{h} \leq 2k_\phi + \overline{h}, \end{aligned}$$

and θ_{k+1} in (3.10) satisfies

$$\theta_{k+1} \geq \underline{\theta} = \frac{(1-\eta)(1-r)\underline{h}}{2k_\phi + \overline{h}}, \tag{3.21}$$

which completes the proof. □

In the following we analyze the possible values taken by $Ared_k(x_{k+1}, \theta_{k+1})$ in case of successful iterations and distinguish the iteration indexes k as below:

$$\mathcal{I}_1 = \{k \geq 0 \text{ s.t. } h(N_k) - h(\widetilde{N}_{k+1}) > 0\}, \tag{3.22}$$

$$\mathcal{I}_2 = \{k \geq 0 \text{ s.t. } h(N_k) = h(\widetilde{N}_{k+1}) = 0, N_{k+1}^t = N\}, \tag{3.23}$$

$$\mathcal{I}_3 = \{k \geq 0 \text{ s.t. } h(N_k) = h(\widetilde{N}_{k+1}) = 0,\ N_{k+1}^t < N\}. \tag{3.24}$$

Note that $\mathcal{I}_1, \mathcal{I}, \mathcal{I}_3$ are disjoint and any iteration index k belongs to exactly one of these subsets.

LEMMA 3.3. *Let Assumptions 3.1 and 3.2 hold and Δ_k be the trust-region radius used at Step 6. Further, suppose that iteration k is successful. If $k \in \mathcal{I}_1$ then*

$$Ared_k(x_{k+1}, \theta_{k+1}) \geq \frac{\eta^2(1-r)\underline{h}}{\Delta_{\max}^2}\Delta_k^2.$$

Otherwise,

$$Ared_k(x_{k+1}, \theta_{k+1}) \geq \eta\eta_2\underline{\theta}\Delta_k^2.$$

Proof. Since iteration k is successful, $x_{k+1} = x_k + p_k$ and (3.16) hold. Suppose $k \in \mathcal{I}_1$. By (3.16) and (3.14)

$$Ared_k(x_k + p_k, \theta_{k+1}) \geq \eta Pred_k(\theta_{k+1}) \geq \eta^2(h(N_k) - h(\widetilde{N}_{k+1})).$$

In virtue of (3.18) we have $h(N_k) - h(\widetilde{N}_{k+1}) \geq (1-r)h(N_k)$, hence we obtain

$$Ared_k(x_k + p_k, \theta_{k+1}) \geq \eta^2(1-r)h(N_k).$$

Dividing and multiplying the previous inequality by Δ_k^2, and applying the inequalities $\underline{h} \leq h(N_k)$, $\Delta_k \leq \Delta_{\max}$, we get the thesis.

Suppose $k \in \mathcal{I}_2$. Then $N_k = N$, $\mathcal{F}_k = 1$ and Step 6 of Algorithm 3.1 enforces $\|g_k\| \geq \eta_2\Delta_k$. By definition of $Pred_k(\theta_{k+1})$ and Lemma 3.2 we have

$$Pred_k(\theta_{k+1}) = \theta_{k+1}(f_N(x_k) - m_k(p_k)) = \theta_{k+1}\Delta_k\|g_k\| \geq \eta_2\underline{\theta}\Delta_k^2,$$

and therefore (3.16) yields the statement.

Suppose $k \in \mathcal{I}_3$. Inequality (3.16) ensures

$$Ared_k(x_k + p_k, \theta_{k+1}) \geq \eta Pred_k(\theta_{k+1}) = \eta\theta_{k+1}(f_N(x_k) - m_k(p_k)).$$

By Step 4 of Algorithm 3.1 and condition $\|g_k\| \geq \eta_2\Delta_k$, enforced in Step 6, we get

$$f_N(x_k) - m_k(p_k) \geq \Delta_k\|g_k\| \geq \eta_2\Delta_k^2,$$

and the statement follows. □

Let us now define a Lyapunov type function Φ similar to the one defined for STORM in [14]. Assumption 3.1 implies that $h(N_k)$ is bounded from above while Assumption 3.2 implies that $f_{N_k}(x)$ is bounded from below if $x \in \Omega$. Thus, there exists a constant Σ such that

$$f_{N_k}(x) - h(N_k) + \Sigma \geq 0, \quad x \in \Omega, \quad k \geq 0. \tag{3.25}$$

Let $v \in (0,1)$ be a fixed constant to be specified below and Δ_k the length of p_k, i.e., the trust-region radius given at the beginning of iteration k when $k \in \mathcal{I}_1 \cup \mathcal{I}_2$ or the trust-region radius at termination of Step 4 otherwise. Then, we introduce

$$\Phi_k \stackrel{\text{def}}{=} \Phi(x_k, N_k, \theta_k, \Delta_k) = v\left(\Psi(x_k, N_k, \theta_k) + \theta_k\Sigma\right) + (1-v)\Delta_k^2, \tag{3.26}$$

where Ψ is the merit function given in (3.3).

First, note that Φ_k is bounded below for all $k \geq 0$,

$$\begin{aligned}\Phi_k &= v\left(\Psi(x_k, N_k, \theta_k) + \theta_k\Sigma\right) + (1-v)\Delta_k^2 \geq v\left(\Psi(x_k, N_k, \theta_k) + \theta_k\Sigma\right) \\ &\geq v\left(\theta_k f_{N_k}(x_k) + (1-\theta_k)h(N_k) + \theta_k(-f_{N_k}(x_k) + h(N_k))\right) \\ &\geq vh(N_k) \geq 0. \end{aligned} \tag{3.27}$$

Second, by the definition (3.26), for all $k > 0$, we have

$$\begin{aligned}\Phi_{k+1} - \Phi_k &= v\left(\theta_{k+1} f_{N_{k+1}}(x_{k+1}) + (1-\theta_{k+1})h(N_{k+1})\right) \\ &\quad -v\left(\theta_k f_{N_k}(x_k) + (1-\theta_k)h(N_k)\right) + v(\theta_{k+1} - \theta_k)\Sigma + (1-v)(\Delta_{k+1}^2 - \Delta_k^2) \\ &= v\left(\theta_{k+1} f_{N_{k+1}}(x_{k+1}) + (1-\theta_{k+1})h(N_{k+1})\right) \pm v\theta_{k+1} f_{N_k}(x_k) \pm v(1-\theta_{k+1})h(N_k) \\ &\quad -v\left(\theta_k f_{N_k}(x_k) + (1-\theta_k)h(N_k)\right) + v(\theta_{k+1} - \theta_k)\Sigma + (1-v)(\Delta_{k+1}^2 - \Delta_k^2) \\ &= v\left(\theta_{k+1}(f_{N_{k+1}}(x_{k+1}) - f_{N_k}(x_k)) + (1-\theta_{k+1})(h(N_{k+1}) - h(N_k))\right) \\ &\quad +v(\theta_{k+1} - \theta_k)(f_{N_k}(x_k) - h(N_k) + \Sigma) + (1-v)(\Delta_{k+1}^2 - \Delta_k^2). \end{aligned} \tag{3.28}$$

If the iteration k is successful, then using (3.25), the monotonicity of $\{\theta_k\}_{k\in\mathbb{N}}$ and the fact that $N_{k+1} = N_{k+1}^t$, the equality (3.28) yields

$$\Phi_{k+1} - \Phi_k \leq -vAred_k(x_{k+1}, \theta_{k+1}) + (1-v)(\Delta_{k+1}^2 - \Delta_k^2). \tag{3.29}$$

Otherwise, if the iteration k is unsuccessful, then $x_{k+1} = x_k$, $N_{k+1} = N_k$ and thus the first quantity at the right-hand side of equality (3.28) is zero. Hence using again (3.25) and the monotonicity of $\{\theta_k\}_{k\in\mathbb{N}}$, we obtain

$$\Phi_{k+1} - \Phi_k \leq (1-v)(\Delta_{k+1}^2 - \Delta_k^2). \tag{3.30}$$

Now we provide bounds for the change of Φ along subsequent iterations and again distinguish the three cases $k \in \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ stated in (3.22)-(3.24).

LEMMA 3.4. *Let Assumptions 3.1–3.2 hold.*

i) If the iteration k is unsuccessful, then

$$\Phi_{k+1} - \Phi_k \leq (1-v)\frac{1-\gamma^2}{\gamma^2}\Delta_k^2. \tag{3.31}$$

ii) If the iteration k is successful and $k \in \mathcal{I}_1$, then

$$\Phi_{k+1} - \Phi_k \leq -v\left(\frac{\eta^2(1-r)\underline{h}}{\Delta_{\max}^2}\right)\Delta_k^2. \tag{3.32}$$

iii) If the iteration k is successful and $k \in \mathcal{I}_2 \cup \mathcal{I}_3$, then

$$\Phi_{k+1} - \Phi_k \leq \left(-v\eta\eta_2\underline{\theta} + (1-v)(\gamma^2 - 1)\right)\Delta_k^2. \tag{3.33}$$

Proof. i) Since $\Delta_{k+1} = \Delta_k/\gamma$ in case of unsuccessful iterations, equation (3.30) directly yields (3.31).

ii) Suppose $k \in \mathcal{I}_1$ is successful and consider Δ_{k+1} in (3.26), namely either the trust-region radius given at the beginning of iteration $k+1$, when $k+1 \in \mathcal{I}_1 \cup \mathcal{I}_2$, or the trust-region radius at termination of Step 4 if $k+1 \in \mathcal{I}_3$. The updating rule (3.11) for Δ_{k+1} and Step 4 imply $\Delta_{k+1} = \Delta_k$ if Step 4 is skipped, $\Delta_{k+1} \leq \Delta_k$ otherwise. Thus combining (3.29) with Lemma 3.3 we obtain (3.32).

iii) Suppose $k \in \mathcal{I}_2 \cup \mathcal{I}_3$ is successful and consider Δ_{k+1} in (3.26), namely either the trust-region radius given at the beginning of iteration $k+1$, when $k+1 \in \mathcal{I}_1 \cup \mathcal{I}_2$, or the trust-region radius at termination of Step 4 if $k+1 \in \mathcal{I}_3$. By the updating rule (3.11), at the beginning of iteration $k+1$ we have $\Delta_{k+1} \leq \gamma \Delta_k$. Thus, repeating the arguments in item ii), we can combine (3.29) with Lemma 3.3 to obtain (3.33). □

We are now ready to prove that a sufficient decrease condition holds for Φ along subsequent iterations.

THEOREM 3.5. *Let Assumptions 3.1–3.2 hold. There exist $v \in (0,1)$ and $\sigma > 0$ such that*

$$\Phi_{k+1} - \Phi_k \leq -\sigma \Delta_k^2, \quad \text{for all } k \geq 0. \tag{3.34}$$

Proof. In case of unsuccessful iterations, equation (3.31) guarantees that the sufficient decrease condition (3.34) holds for any value of $v \in (0,1)$.
Let us now consider successful iterations. If $k \in \mathcal{I}_1$, condition (3.34) is guaranteed by (3.32) for $v \in (0,1)$. Otherwise, if $k \in \mathcal{I}_2 \cup \mathcal{I}_3$, then the sufficient decrease holds if and only if the scalar multiplying Δ_k^2 in (3.33) is negative, namely

$$\frac{\gamma^2 - 1}{\eta \eta_2 \underline{\theta} + \gamma^2 - 1} < v < 1. \tag{3.35}$$

Therefore, if v is chosen as to satisfy (3.35) then (3.31)-(3.33) imply (3.34) with

$$\sigma = \min \left\{ (1-v)\frac{\gamma^2 - 1}{\gamma^2}, \frac{v\eta^2(1-r)\underline{h}}{\Delta_{\max}^2}, v\eta\eta_2\underline{\theta} + (1-v)(1-\gamma^2) \right\}.$$

□

In order to proceed in our analysis, we make the following two assumptions.

ASSUMPTION 3.3. *For all $i = 1, \ldots, N$, the functions ϕ_i are continuous and the gradients $\nabla \phi_i$ are Lipschitz continuous on Ω with constant L_i. Let $L = \frac{1}{2} \max_{1 \leq i \leq N} L_i$.*

ASSUMPTION 3.4. *Assume that there exists a positive scalar Γ such that*

$$\left\| g_k - \nabla f_{N_{k+1}^t}(x_k) \right\| \leq \Gamma$$

for all x_k, g_k and N_{k+1}^t generated by Algorithm 3.1.

REMARK 3.6. Assumption 3.4 is satisfied if Assumptions 3.2 and 3.3 hold with $\Omega = \mathbb{R}^n$ and the direction g_k is a subsampled gradient of the form

$$g_k = \nabla f_{N_{k+1,g}}(x_k) = \frac{1}{N_{k+1,g}} \sum_{i \in I_{N_{k+1,g}}} \nabla \phi_i(x_k), \tag{3.36}$$

where $I_{N_{k+1,g}} \subseteq I_{N_{k+1}^t}$ and $N_{k+1,g} = |I_{N_{k+1,g}}|$. In fact, we can write

$$\left\| g_k - \nabla f_{N_{k+1}^t}(x_k) \right\| \leq \|\nabla f_{N_{k+1,g}}(x_k)\| + \|\nabla f_{N_{k+1}^t}(x_k)\|, \tag{3.37}$$

and derive an upper bound for the norm of ∇f_M with M integer, $1 \leq M \leq N$. Setting $L = \frac{1}{2} \max_{1 \leq i \leq N} L_i$, it is easy to show that ∇f_M is Lipschitz continuous on $\mathbb{R}^n$ with

constant $2L$. Then Assumption 3.2 and the descent lemma for continuously differentiable functions with Lipschitz continuous gradient [6, Proposition A.24] ensures

$$f_{low} \leq f_M(y) \leq f_M(x) + \nabla f_M(x)^T(y-x) + L\|y-x\|^2, \quad \forall\, x, y \in \mathbb{R}^n.$$

Taking the minimum of the right-hand side with respect to y, we can also write

$$f_{low} \leq \min_{y\in\mathbb{R}^n} \zeta(y) \equiv f_M(x) + \nabla f_M(x)^T(y-x) + L\|y-x\|^2, \quad \forall\, x \in \mathbb{R}^n.$$

The minimum of $\zeta(y)$ is attained at the point $\bar{y} = x - \frac{1}{2L}\nabla f_M(x)$ and letting $y = \bar{y}$ in the previous inequality, we get:

$$f_{low} \leq f_M(x) - \frac{1}{2L}\|\nabla f_M(x)\|^2 + \frac{1}{4L}\|\nabla f_M(x)\|^2, \quad \forall\, x \in \mathbb{R}^n.$$

and equivalently $\|\nabla f_M(x)\|^2 \leq 4L(f_M(x) - f_{low}), \quad \forall\, x \in \mathbb{R}^n$. Using again Assumption 3.2, we have $f_M(x) - f_{low} \leq |f_M(x)| + |f_{low}| \leq 2\kappa_\phi$, and consequently

$$\|\nabla f_M(x)\|^2 \leq 8L\kappa_\phi, \quad \forall\, x \in \mathbb{R}^n.$$

Thus, (3.37) gives Assumption 3.4 with $\Gamma = 4\sqrt{2L\kappa_\phi}$

The following estimate will be useful in further analysis.

LEMMA 3.7. *Suppose that Assumptions 3.3–3.4 hold. Then*

$$|m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)| \leq (\Gamma + L\Delta_k)\Delta_k. \tag{3.38}$$

Proof. From Assumption 3.3, it follows that $\nabla f_{N^t_{k+1}}$ is Lipschitz continuous with constant $2L$. We note that

$$\begin{aligned} |m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)| &= \left| \int_0^1 \left(g_k \pm \nabla f_{N^t_{k+1}}(x_k) - \nabla f_{N^t_{k+1}}(x_k + \tau p_k) \right)^T p_k d\tau \right| \\ &\leq \int_0^1 \|g_k - \nabla f_{N^t_{k+1}}(x_k)\|\|p_k\| d\tau + \int_0^1 2L\tau\|p_k\|^2 d\tau. \end{aligned} \tag{3.39}$$

Then, Assumption 3.4 and the equality $\|p_k\| = \Delta_k$ give the thesis. □

The next result shows that a successful iteration in $\mathcal{I}_1$ is guaranteed when the trust-region radius is small enough. Let us introduce the following constant

$$\delta_2 = \eta(1-\eta)(1-r)\underline{h}. \tag{3.40}$$

LEMMA 3.8. *Let Assumptions 3.1–3.4 hold and suppose that $k \in \mathcal{I}_1$ with $\mathcal{I}_1$ defined in (3.22). Then the iteration is successful whenever $\Delta_k \leq \Delta$, where*

$$\Delta = \frac{-\Gamma + \sqrt{\Gamma^2 + 4(\mu + L)\delta_2}}{2(\mu + L)}. \tag{3.41}$$

Proof. By (3.13), (3.15) and (3.14) we obtain

$$\begin{aligned} Ared_k(x_k + p_k, \theta_{k+1}) - \eta Pred_k(\theta_{k+1}) &= (1-\eta)\mathrm{Pred}_k(\theta_{k+1}) \\ &\quad + \theta_{k+1}(m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)) + (1-\theta_{k+1})(h(\widetilde{N}_{k+1}) - h(N^t_{k+1})) \\ &\geq \eta(1-\eta)(h(N_k) - h(\widetilde{N}_{k+1})) + \theta_{k+1}(m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)) \\ &\quad + (1-\theta_{k+1})(h(\widetilde{N}_{k+1}) - h(N^t_{k+1})). \end{aligned} \tag{3.42}$$

Using (3.38), (3.8) and $\theta_{k+1} \leq \theta_0 \leq 1$, we also have

$$|\theta_{k+1}(m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)) + (1 - \theta_{k+1})(h(\widetilde{N}_{k+1}) - h(N^t_{k+1}))| \\ \leq \Gamma\Delta_k + (\mu + L)\Delta_k^2. \tag{3.43}$$

Since (3.18) guarantees that $\eta(1-\eta)(h(N_k) - h(\widetilde{N}_{k+1})) \geq \delta_2$, from (3.42) and (3.43) we have

$$Ared_k(x_k + p_k, \theta_{k+1}) - \eta Pred_k(\theta_{k+1}) \geq \delta_2 - \Gamma\Delta_k - (\mu + L)\Delta_k^2.$$

Then, analysing the quadratic function at the right hand side of the above inequality, we conclude that the iteration is successful whenever $\Delta_k \leq \Delta$. □

We conclude the first part of our analysis showing that eventually $N_k = N^t_{k+1} = \widetilde{N}_{k+1} = N$.

THEOREM 3.9. *Let Assumptions 3.1 – 3.4 hold. Then the sequence* $\{\Delta_k\}$ *in Algorithm 3.1 satisfies*

$$\lim_{k\to\infty} \Delta_k = 0,$$

and $N_k = N^t_{k+1} = \widetilde{N}_{k+1} = N$ *for* k *sufficiently large.*

Proof. Under the stated conditions Theorem 3.5 holds and summing up (3.34) for $j = 0, 1, \ldots, k-1$, we obtain

$$\Phi_k - \Phi_0 = \sum_{j=0}^{k-1} (\Phi_{j+1} - \Phi_j) \leq -\sigma \sum_{j=0}^{k-1} \Delta_j^2.$$

Given that, by (3.27), Φ_k is bounded from below for all k, we conclude that $\sum_{j=0}^{\infty} \Delta_j^2 < \infty$, and hence $\lim_{j\to\infty} \Delta_j = 0$. As a consequence we have

$$\lim_{k\to\infty} \Big(h(N^t_{k+1}) - h(\widetilde{N}_{k+1}) \Big) = 0,$$

due to (3.8). Therefore, for k large enough we have $N^t_{k+1} = \widetilde{N}_{k+1}$. Indeed, if $N^t_{k+1} < \widetilde{N}_{k+1}$ occurs for k arbitrarily large, then $h(N^t_{k+1}) - h(\widetilde{N}_{k+1}) \geq \min\{h(M_1) - h(M_2), 1 \leq M_1 < M_2 \leq N\} > 0$ by Assumption 3.1. This contradicts that $h(N^t_{k+1}) - h(\widetilde{N}_{k+1})$ tends to zero.
Now let $k_1 \geq 0$ be the iteration such that the following two conditions hold

$$N^t_{k+1} = \widetilde{N}_{k+1}, \quad \Delta_k \leq \Delta, \quad \text{for all } k \geq k_1,$$

where Δ is defined as in (3.41). If all $k \geq k_1$ belonged to $\mathcal{I}_1$, then the iterations would always be successful thanks to Lemma 3.8 and, because of the update rule (3.11), we would have Δ_k constant and not converging to zero. Then there must exist some $\ell \geq k_1$ such that $\ell \in \mathcal{I}_2 \cup \mathcal{I}_3$ and thus $N_\ell = \widetilde{N}_{\ell+1} = N$. Since $N^t_{k+1} = \widetilde{N}_{k+1}$ for $k \geq \ell$, Step 6 of Algorithm 3.1 implies that $k \in \mathcal{I}_2$ for all $k \geq \ell$, namely $N_k = N^t_{k+1} = \widetilde{N}_{k+1} = N$ for all k sufficiently large. □

Theorem 3.9 allows us to conclude that, after a sufficiently large number of iterations, Algorithm 3.1 reduces to the trust-region method with random gradients and

exact function evaluations first presented in [1] and later extended to include random function estimates in [14] under the name of STORM. As a result, Algorithm 3.1 shares the same lim-type convergence result in probability deduced for the method in [1], provided that the model $m_k(p)$ is sufficiently accurate in probability.

THEOREM 3.10. *Let Assumptions 3.1 – 3.4 hold and $\{x_k\}$ be generated by Algorithm 3.1. Let ℓ be the index such that $N_k = \widetilde{N}_{k+1} = N_{k+1}^t = N$, $\forall k \geq \ell$, and suppose that* (2.2) *holds with probability $p_M \geq \frac{1}{2}$, $\forall k \geq \ell$. Then almost surely we have*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Proof. The proof follows observing that m_k reduces to ς_k in (2.2) for $\forall k \geq \ell$ and by combining Theorem 3.9 with [1, Theorem 4.3]. □

3.2. Iteration complexity. Theorem 3.9 implies that there exists $\ell > 0$ such that our Algorithm 3.1 reduces, for $k \geq \ell$, to the STORM algorithm with exact function values f_N and random gradients. Specializing complexity results provided in [9], it follows that, if (2.2) holds with probability $p_M \in (1/2, 1)$ sufficiently large, then the expected number of iterations required by STORM to achieve $\|\nabla f_N(x)\| \leq \epsilon$ in the worst case is of the order of $O\left(\frac{p_M \epsilon^{-2}}{2p_M - 1}\right)$. Therefore, we can conclude that our procedure needs in expectation at most $O\left(\frac{p_M \epsilon^{-2}}{2p_M - 1} + \ell\right)$ iterations in the worst case to reach $\|\nabla f_N(x)\| \leq \epsilon$.

The aim of this subsection is to provide an upper bound for ℓ. To this end, we denote with $\mathcal{I}_2^\ell$ the subset of indices of $\mathcal{I}_2$ defined in (3.23) before SIRTR definitively reduces to STORM:

$$\mathcal{I}_2^\ell = \{k \in \mathcal{I}_2 : k < \ell\}. \tag{3.44}$$

Then the number of iterations ℓ is the cardinality of the union of sets $\mathcal{I}_1$, $\mathcal{I}_2^\ell$, $\mathcal{I}_3$ defined in (3.22), (3.44), (3.24). Furthermore we observe that, whenever $h(\widetilde{N}_{k+1}) = 0$ and $\Delta_k^2 < \underline{h}/\mu$, condition (3.8) yields $h(N_{k+1}^t) = 0$ and Step 6 implies $\Delta_{k+1} \leq \Delta_k < \sqrt{\underline{h}/\mu}$. Then, given an iteration $k \in \mathcal{I}_2$ such that $\Delta_k^2 < \underline{h}/\mu$, we can conclude that any subsequent iteration belongs to $\mathcal{I}_2$, namely $k \geq \ell$. Then defining the following set of indices

$$\mathcal{I}_2^{\underline{h},\mu} = \left\{k \in \mathcal{I}_2 :\ \Delta_k^2 \geq \underline{h}/\mu\right\}, \tag{3.45}$$

the number of iterations ℓ can be upper bounded as follows

$$\ell \leq |\mathcal{I}_1 \cup \mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3|. \tag{3.46}$$

We now wish to estimate the cardinality $|\mathcal{I}_1 \cup \mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3|$, in order to provide the desired upper bound for ℓ. To this aim, we introduce the following sets of indices

$$\mathcal{I}_{1,S} = \{j \in \mathcal{I}_1,\ \text{iteration } j \text{ is successful}\} \tag{3.47}$$

$$\mathcal{I}_{2,S}^{\underline{h},\mu} = \{j \in \mathcal{I}_2^{\underline{h},\mu},\ \text{iteration } j \text{ is successful}\} \tag{3.48}$$

$$\mathcal{I}_{3,S} = \{j \in \mathcal{I}_3,\ \text{iteration } j \text{ is successful}\} \tag{3.49}$$

$$\mathcal{I}_U = \{j \in \mathcal{I}_1 \cup \mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3,\ \text{iteration } j \text{ is unsuccessful}\} \tag{3.50}$$

$$\mathcal{I}_S = \{j \in \mathcal{I}_1 \cup \mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3,\ \text{iteration } j \text{ is successful}\} \tag{3.51}$$

and estimate their maximum possible cardinalities.

We note that the availability of accurate gradients, in the sense that the event

$$\mathcal{G}_k = \{g_k \in \mathbb{R}^n : \|g_k - \nabla f_{N^t_{k+1}}(x_k)\| \leq K\Delta_k\} \tag{3.52}$$

is true, has an impact on the acceptance of the trial steps whenever $k \in \mathcal{I}_1$. The following lemma establishes a threshold $\overline{\Delta}$ such that if (3.52) is satisfied and $\Delta_k \leq \overline{\Delta}$, then the iteration is successful.

LEMMA 3.11. *Let Assumptions 3.1–3.4 hold, and* $L = \frac{1}{2}\max_{1\leq i\leq N} L_i$. *Suppose that* $k \in \mathcal{I}_1$ *with* $\mathcal{I}_1$ *defined in (3.22) and that the event* $\mathcal{G}_k$ *in (3.52) is true. Then, iteration* k *is successful whenever* $\Delta_k \leq \overline{\Delta}$, *with*

$$\overline{\Delta} = \sqrt{\frac{\delta_2}{K + L + \mu}}, \tag{3.53}$$

and δ_2 *given by (3.40).*

Proof. By (3.39) and (3.52) it follows

$$|m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)| \leq (K + L)\Delta_k^2,$$

where $2L$ is the Lipschitz constant of ∇f. Thus, repeating the arguments of Lemma 3.8, the claim easily follows. □

REMARK 3.12. *Considering Lemma 3.8 and Lemma 3.11, notice that by (3.41) and (3.53) we have*

$$\Delta = \frac{\sqrt{\delta_2}}{\frac{\Gamma}{2\sqrt{\delta_2}} + \sqrt{\frac{\Gamma^2}{4\delta_2} + \mu + L}} \leq \frac{\sqrt{\delta_2}}{\sqrt{\frac{\Gamma^2}{4\delta_2} + L + \mu}} = \overline{\Delta}\,\frac{\sqrt{K + L + \mu}}{\sqrt{\frac{\Gamma^2}{4\delta_2} + L + \mu}}.$$

Thus, $\overline{\Delta} > \Delta$ *whenever* $K < \frac{\Gamma^2}{4\delta_2}$. *Namely, if* K *in (3.53) is moderate and the gradient is accurate, the iteration is successful for larger trust-region radius than in the case where we can use only Assumption 3.4.*

The main result on iteration complexity is given below. We denote the indicator of the random event $\mathcal{G}$ occurring by $\mathbb{1}_{\mathcal{G}}$ while the notation $\mathcal{G}^c$ indicates that event $\mathcal{G}$ is not occurring.

THEOREM 3.13. *Assume that Assumptions 3.1–3.4 hold. Let* $\gamma > 1$, $\mu > 0$, η, $r \in (0, 1)$, $\eta_2 > 0$, $0 < \Delta_0 < \Delta_{\max}$ *be the parameters appearing in Algorithm 3.1,* v *the constant in Theorem 3.5 satisfying* (3.35), Δ *and* $\overline{\Delta}$ *given in* (3.41) *and (3.53), respectively,* $\underline{h} > 0$ *defined as in* (3.2), *and*

$$\widetilde{h} \stackrel{\text{def}}{=} \min\{h(M_1) - h(M_2), 1 \leq M_1 < M_2 \leq N\} > 0.$$

i) The cardinality of $\mathcal{I}_{1,S}$ *in* (3.47) *satisfies*

$$|\mathcal{I}_{1,S}| \leq \left\lceil \frac{\Phi_0}{\widetilde{h}v\eta^2} \right\rceil. \tag{3.54}$$

ii) If $\Delta_0 < \sqrt{\underline{h}/\mu}$, *then the set* $\mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3$ *defined in* (3.24)-(3.44) *is empty, otherwise its cardinality satisfies*

$$|\mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}| \leq \left\lceil \frac{\mu\Phi_0}{\underline{h}(v\eta\eta_2\underline{\theta} - (1 - v)(\gamma^2 - 1))} \right\rceil. \tag{3.55}$$

iii) If $\Delta_0 < \min\{\Delta, \overline{\Delta}, \sqrt{\underline{h}/\mu}\}$*, then the set* $\mathcal{I}_U$ *in* (3.50) *is empty, otherwise its cardinality satisfies*

$$|\mathcal{I}_U| \leq \left\lceil \sum_{j \in \mathcal{I}_S} \mathbb{1}_{\mathcal{G}_j^c} \left(\log_\gamma \left(\frac{\Delta_{\max}}{\min\{\Delta, \sqrt{h/\mu}\}} \right) + 1 \right) + \mathbb{1}_{\mathcal{G}_j} \left(\log_\gamma \left(\frac{\Delta_{\max}}{\min\{\overline{\Delta}, \sqrt{h/\mu}\}} \right) + 1 \right) \right\rceil, \tag{3.56}$$

where $\mathcal{I}_S$ *is given in* (3.51).

Proof. i) If the iteration k is successful, equations (3.7) and (3.14) give

$$h(N_k) - h(\widetilde{N}_{k+1}) \leq \frac{\mathrm{Pred}_k(\theta_{k+1})}{\eta} \leq \frac{\mathrm{Ared}_k(x_{k+1}, \theta_{k+1})}{\eta^2}.$$

Consequently, recalling (3.47) and using (3.29) and $\Delta_{k+1} = \Delta_k$ whenever $k \in \mathcal{I}_{1,S}$, we get

$$\sum_{k \in \mathcal{I}_{1,S}} (h(N_k) - h(\widetilde{N}_{k+1})) \leq \frac{1}{v\eta^2} \sum_{k \in \mathcal{I}_{1,S}} (\Phi_k - \Phi_{k+1}). \tag{3.57}$$

For any $k \in \mathcal{I}_{1,S}$, we have $0 \leq k < \ell$, and $\Phi_k - \Phi_{k+1} \geq 0$ by Theorem 3.5. Hence we can write

$$\sum_{k \in \mathcal{I}_{1,S}} (\Phi_k - \Phi_{k+1}) \leq \sum_{k=0}^{\ell-1} (\Phi_k - \Phi_{k+1}) = \Phi_0 - \Phi_\ell \leq \Phi_0,$$

where the last inequality follows from (3.27). Then (3.57) yields

$$\sum_{k \in \mathcal{I}_{1,S}} (h(N_k) - h(\widetilde{N}_{k+1})) \leq \frac{\Phi_0}{v\eta^2}.$$

Finally, since $N_k < \widetilde{N}_{k+1} \leq N$ whenever $k \in \mathcal{I}_{1,S}$, it follows $h(N_k) - h(\widetilde{N}_{k+1}) \geq \widetilde{h}$ for any $k \in \mathcal{I}_{1,S}$, we conclude that the number of indices in $\mathcal{I}_{1,S}$ is bounded from above as in (3.54).

ii) In case $\Delta_0 < \sqrt{\underline{h}/\mu}$, since the updating rule in Step 6 allows to increase the trust-region radius only when $k \in \mathcal{I}_2$ and $\Delta_k^2 \geq \underline{h}/\mu$, it follows that $\Delta_k^2 < \underline{h}/\mu$ for $k = 0, 1, \ldots, \ell$ and thus the set $\mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3$ is empty.

Let us consider the case $\Delta_0 \geq \sqrt{\underline{h}/\mu}$. If $k \in \mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}$, then the decrease condition (3.33) holds. Hence we can sum (3.33) over $k \in \mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}$ and obtain

$$\begin{aligned}(v\eta\eta_2\underline{\theta} - (1-v)(\gamma^2 - 1)) \sum_{k \in \mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}} \Delta_k^2 &\leq \sum_{k \in \mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}} (\Phi_k - \Phi_{k+1}) \\ &\leq \sum_{k=0}^{\ell-1} (\Phi_k - \Phi_{k+1}) = \Phi_0 - \Phi_\ell \leq \Phi_0, \end{aligned} \tag{3.58}$$

where the second inequality follows from $\mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S} \subseteq \{0, \ldots, \ell - 1\}$ and Theorem 3.5, whereas the last inequality is obtained using (3.27). Now observe that if $k \in \mathcal{I}_{2,S}^{\underline{h},\mu}$ then $\Delta_k^2 \geq \underline{h}/\mu$ by definition of $\mathcal{I}_2^{\underline{h},\mu}$; if $k \in \mathcal{I}_{3,S}$ then $h(\widetilde{N}_{k+1}) = 0$, $h(N_{k+1}^t) \geq \underline{h}$ and condition (3.9) implies again $\Delta_k^2 \geq \underline{h}/\mu$. Since the constant multiplying Δ_k^2 in (3.58)

is positive thanks to (3.35), we can combine (3.58) with the lower bound $\Delta_k^2 \geq \underline{h}/\mu$ to get the claim.

iii) If $\Delta_k < \min\{\Delta, \overline{\Delta}, \sqrt{\underline{h}/\mu}\}$, by Lemma 3.8) any iteration in $\mathcal{I}_1$ is successful. Moreover, as we observed in Item ii), k does not belong to the set $\mathcal{I}_2^{\underline{h},\mu} \cup \mathcal{I}_3$. Thus, if $\Delta_0 < \min\{\Delta, \overline{\Delta}, \sqrt{\underline{h}/\mu}\}$, the updating rule in Step 6 implies that the set $\mathcal{I}_U$ in (3.50) is empty. Otherwise, let $t \geq 1$ be the number of unsuccessful iterations between two successive successful iterations $(k-t-1)$ and $k < \ell$. If $\mathbb{1}_{\mathcal{G}_j^c} = 1$, the updating rule for unsuccessful iterations in Step 6 implies

$$\min(\Delta, \sqrt{\underline{h}/\mu}) \leq \Delta_{k-1} = \frac{\Delta_{k-2}}{\gamma} = \frac{\Delta_{k-t}}{\gamma^{t-1}} \leq \frac{\Delta_{\max}}{\gamma^{t-1}}.$$

Therefore, the number t of unsuccessful iterations between two successive successful iterations is at most

$$t \leq \log_\gamma \left(\frac{\Delta_{\max}}{\min(\Delta, \sqrt{\underline{h}/\mu})} \right) + 1. \tag{3.59}$$

Applying the same reasoning to the case where $\mathbb{1}_{\mathcal{G}_j} = 1$, $\Delta_k < \min\{\overline{\Delta}, \sqrt{\underline{h}/\mu}\}$(3.56) follows. □

Let us analyze the previous estimates. First, setting $p_g \in (0,1]$ as the probability for event (3.52), we conclude that the expected value $E(|\mathcal{I}_U|)$ is

$$E(|\mathcal{I}_U|) \leq |\mathcal{I}_S| \left\lceil (1-p_g) \left(\log_\gamma \left(\frac{\Delta_{\max}}{\min\{\Delta, \sqrt{h/\mu}\}} \right) + 1 \right) + p_g \left(\log_\gamma \left(\frac{\Delta_{\max}}{\min\{\overline{\Delta}, \sqrt{h/\mu}\}} \right) + 1 \right) \right\rceil .$$

Second, consider the special setting where the infeasibility measure is defined by $h(M) = (N-M)/N$ (thus $\underline{h} = \tilde{h} = 1/N$), $\mu = O(1/N)$, and

$$v = \frac{\gamma^2 - 1 + \alpha\eta\eta_2\underline{\theta}}{\gamma^2 - 1 + \eta\eta_2\underline{\theta}}, \quad \alpha \in (0,1). \tag{3.60}$$

Note that (3.60) complies with (3.35). We further note that $\alpha \in (0,1)$ can be chosen arbitrarily close to 1, as well as v and that the parameter $\underline{\theta}$ is directly proportional to $\underline{h} = 1/N$, by (3.21). From (3.54) we immediately get

$$|\mathcal{I}_{1,S}| \leq O\left(\frac{N}{\eta^2}\right).$$

As for (3.55), v in (3.60) gives

$$\frac{\mu\Phi_0}{\underline{h}(v\eta\eta_2\underline{\theta} - (1-v)(\gamma^2-1))} = \frac{\mu\Phi_0}{\alpha\eta\eta_2\underline{h}\ \underline{\theta}},$$

and equation (3.55) implies

$$|\mathcal{I}_{2,S}^{\underline{h},\mu} \cup \mathcal{I}_{3,S}| \leq O\left(\frac{N}{\eta\eta_2}\right).$$

From equations (3.40)-(3.41) and (3.53) we also deduce that $\Delta = O(1/N)$ and $\bar{\Delta} = O(1/\sqrt{N})$, and combining the previous remarks with (3.56) we get

$$|\mathcal{I}_U| \leq O\left(\frac{N\ \log_\gamma N}{\eta^2\eta_2}\right).$$

	Training set		Testing set
Data set	N	n	N_T
CINA0	10000	132	6033
A9A	22793	123	9768
COVERTYPE	464810	54	116202
IJCNN1	49990	22	91701
MNIST	60000	784	10000
HTRU2	10000	8	7898

TABLE 4.1
Data sets used

Therefore, in this special setting, we can say that the number of iterations performed before SIRTR reduces turns into STORM is at most of order $O((N \log_\gamma N)/(\eta^2\eta_2))$.

We conclude this section noting that it is easy to see that when f_N is Lipschitz continuous, see Assumption 3.3, and $N_{k+1}^t = N$, then (2.2) holds at least with probability $p_g \in (\frac{1}{2}, 1)$ whenever the event (3.52) holds with the same probability. Therefore, under this latter assumption and p_g sufficiently large, Theorem 3.10 ensures $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ and in the worst-case Algorithm 3.1 needs in expectation at most $O\left(\frac{p_M \epsilon^{-2}}{2p_M - 1} + \frac{N \log_\gamma N}{\eta^2\eta_2}\right)$ iterations to reach $\|\nabla f_N(x)\| \leq \epsilon$.

4. Numerical experience. In this section, we evaluate the numerical performance of SIRTR on a nonconvex optimization problem arising in binary classification. Let $\{(a_i, b_i)\}_{i=1}^N$ denote the pairs forming a training set with $a_i \in \mathbb{R}^n$ containing the entries of the i-th example, and $b_i \in \{0, 1\}$ representing the corresponding label. Then, we address the following minimization problem

$$\min_{x\in\mathbb{R}^n} f_N(x) = \frac{1}{N}\sum_{i=1}^{N}\left(b_i - \frac{1}{1+e^{-a_i^T x}}\right)^2, \tag{4.1}$$

where the nonconvex objective function f_N is obtained by composing a least-squares loss with the sigmoid function.
In Table 4.1, we report the information related to the datasets employed, including the number N of training examples, the dimension n of each example and the dimension N_T of the testing set I_{N_T}. All the numerical results have been obtained by running MATLAB R2019a on an Intel Core i7-4510U CPU 2.00-2.60 GHz with an 8 GB RAM.

For all our numerical tests, we set $x_0 = (0, 0, \ldots, 0)^T$ as initial guess, $\Delta_0 = 1$ as the initial trust-region radius, $\Delta_{\max} = 100$, $\gamma = 2$, $\eta = 10^{-1}$, $\eta_2 = 10^{-6}$.

Concerning the inexact restoration phase, we borrow the implementation details from [3]. Specifically, the infeasibility measure h and the initial penalty parameter θ_0 are set as follows:

$$h(M) = \frac{N - M}{N}, \quad \theta_0 = 0.9.$$

The updating rule for choosing $\widetilde{N}_{k+1}$ has the form

$$\widetilde{N}_{k+1} = \min\{N, \lceil \widetilde{c} N_k \rceil\}, \tag{4.2}$$

where $1 < \widetilde{c} < 2$ is a prefixed constant factor; note that this choice of $\widetilde{N}_{k+1}$ satisfies (3.7) with $r = (N - (\widetilde{c} - 1))/N$. At Step 2 the function sample size N_{k+1}^t is computed

using the rule

$$N_{k+1}^t = \begin{cases} \lceil \widetilde{N}_{k+1} - \mu N \Delta_k^2 \rceil, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \Delta_k^2 \rceil \in [N_0, 0.95N] \\ \widetilde{N}_{k+1}, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \Delta_k^2 \rceil < N_0 \\ N, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \Delta_k^2 \rceil > 0.95N. \end{cases} \tag{4.3}$$

Once the set $I_{N_{k+1}^t}$ is fixed, the search direction $g_k \in \mathbb{R}^n$ is computed via sampling as in (3.36) and the sample size $N_{k+1,g}$ is fixed as

$$N_{k+1,g} = \lceil c N_{k+1}^t \rceil, \tag{4.4}$$

with $c \in (0, 1]$ and $I_{N_{k+1,g}} \subseteq I_{N_{k+1}^t}$.

4.1. SIRTR performance. In the following, we show the numerical behaviour of SIRTR and focus on three aspects: the classification error provided by the final iterate, the computational cost, the occurrence of termination before full accuracy in function evaluations is reached. The latter issue is crucial because it indicates the ability of our approach to solve (4.1) with random models and both sampling and steplength selection ruled by the inexact restoration approach.

The average classification error provided by the final iterate, say x_f, is defined as

$$\texttt{err} = \frac{1}{N_T} \sum_{i \in I_{N_T}} |b_i - b_i^{pred}|, \tag{4.5}$$

where b_i is the exact label of the $i-$th instance of the testing set, and b_i^{pred} is the corresponding predicted label, given by $b_i^{pred} = \max\{\text{sign}(a_i^T x_f), 0\}$.

The computational cost is measured in terms of full function and gradient evaluations. In our test problems, the main cost in the computation of ϕ_i, $1 \le i \le N$, is the scalar product $a_i^T x$: once this product is evaluated, it can be reused for computing $\nabla \phi_i$. Nonetheless, following [29, Section 3.3], we count both function and gradient evaluations as if we were addressing a classification problem based on a neural net. Thus, computing a single function ϕ_i requires $\frac{1}{N}$ forward propagations, whereas the gradient evaluation corresponds to $\frac{2}{N}$ propagations (an additional backward propagation is needed). In our implementation, once ϕ_i is computed, the corresponding gradient $\nabla \phi_i$ requires only $\frac{1}{N}$ backward propagations, hence the computational cost of SIRTR at each iteration k is determined by $\frac{N_{k+1} + N_{k+1,g}}{N}$ propagations.

For all experiments in this section, we run SIRTR and stop it when either a maximum of 1000 iterations is reached, a maximum of 500 full function evaluations is performed, or when one of the following two conditions is satisfied

$$\|g_k\| \le \epsilon, \qquad |f_{N_k}(x_k) - f_{N_{k-1}}(x_{k-1})| \le \epsilon |f_{N_{k-1}}(x_{k-1})| + \epsilon, \tag{4.6}$$

where $\epsilon = 10^{-3}$.

Since the selection of sets $I_{N_{k+1}^t}$ and $I_{N_{k+1,g}}$ for computing $f_{N_{k+1}^t}(x_k)$ and g_k is random, we performed 50 runs of SIRTR for each test problem. Results are reported in tables where the headings of the columns have the following meaning: `cost` is the overall number of full function and gradient evaluations averaged over the 50 runs, `err` is the classification error given in (4.5) averaged over the 50 runs, `sub` the number of runs where the method is stopped before reaching full accuracy in function evaluations.

$N_{k+1,g}$	$\lceil 0.1N^t_{k+1} \rceil$			$\lceil 0.2N^t_{k+1} \rceil$			N^t_{k+1}		
	cost	err	sub	cost	err	sub	cost	err	sub
CINA0	10	0.236	46	21	0.217	42	28	0.226	44
A9A	7	0.177	47	6	0.178	50	10	0.179	48
COVERTYPE	7	0.448	48	9	0.433	37	20	0.435	22
IJCNN1	14	0.091	20	13	0.091	23	20	0.089	18
MNIST	7	0.171	47	8	0.173	48	8	0.176	50
HTRU2	8	0.031	46	9	0.029	45	13	0.031	48

TABLE 4.2

Results with three different rules for computing the sample size $N_{k+1,g}$.

$\tilde{N}_{k+1}$	$\min\{N, \lceil 1.05N_k \rceil\}$			$\min\{N, \lceil 1.1N_k \rceil\}$			$\min\{N, \lceil 1.2N_k \rceil\}$		
	cost	err	sub	cost	err	sub	cost	err	sub
CINA0	7	0.246	50	8	0.239	49	10	0.236	46
A9A	4	0.180	50	4	0.178	50	7	0.177	47
COVERTYPE	4	0.418	50	4	0.430	50	7	0.448	48
IJCNN1	4	0.095	50	4	0.095	50	14	0.091	20
MNIST	4	0.164	50	5	0.167	50	7	0.171	47
HTRU2	5	0.030	50	6	0.031	50	8	0.031	46

TABLE 4.3

Results with three different rules for computing the sample size $\widetilde{N}_{k+1}$.

In a first set of experiments, we investigated the choice of $N_{k+1,g}$ by varying the factor $c \in (0, 1]$ in (4.4). In particular, letting $\widetilde{c} = 1.2$ in (4.2), $\mu = 100/N$ in (4.3) and $N_0 = \lceil 0.1N \rceil$ as in [3], we tested the values $c \in \{0.1, 0.2, 1\}$. The results obtained are reported in Table 4.2. We note that the classification error slightly varies with respect to the choice of $N_{k+1,g}$, and that selecting $N_{k+1,g}$ as a small fraction of N^t_{k+1} is convenient from a computationally point of view. By contrast, the choice $N_{k+1,g} = N^t_{k+1}$ leads to the largest computational costs without providing a significant gain in accuracy; besides the cost per iteration, equal to $\frac{2N^t_{k+1}}{N}$, we observe that full accuracy in function evaluations is reached in several runs of COVERTYPE and IJCNN1. In the runs where $N_{k+1,g} = N^t_{k+1}$ and the function is evaluated at full accuracy, the trust-region model is the same as in standard trust-region algorithms. Remarkably, the results in Table 4.2 highlight that random models compare favourably with respect to cost and classification errors.

Next, we show that SIRTR computational cost can be reduced by slowing down the growth rate of N^t_{k+1}. This task can be achieved controlling the growth of $\widetilde{N}_{k+1}$ which affects N^t_{k+1} by means of (4.3). Letting $c = 0.1$, $\mu = 100/N$ and $N_0 = \lceil 0.1N \rceil$, we consider the choices $\widetilde{c} \in \{1.05, 1.1, 1.2\}$ in (4.2). Average results are reported in Table 4.3. We can observe that a slower growth rate for $\widetilde{N}_{k+1}$ gives rise to a reduction of the number of function/gradient evaluations for all datasets, while retaining a similar classification error. Moreover, significantly for $\widetilde{c} = 1.05$ all runs performed stopped before reaching full function accuracy, and for $\widetilde{c} = 1.1$ full accuracy was reached only once out of 50 runs in CINA0 dataset.

We now analyze three different values, $N_0 \in \{\lceil 0.001N \rceil, \lceil 0.01N \rceil, \lceil 0.1N \rceil\}$, for the initial sample size N_0. We apply SIRTR with $\tilde{c} = 1.05$ in (4.2), $\mu = 100/N$ in (4.3), and $c = 0.1$ in (4.4). Results are reported in Table 4.4. We can see that,

N_0	$\lceil 0.001N \rceil$			$\lceil 0.01N \rceil$			$\lceil 0.1N \rceil$		
	cost	err	sub	cost	err	sub	cost	err	sub
CINA0	2	0.230	50	4	0.215	49	7	0.246	50
A9A	2	0.217	50	4	0.180	50	6	0.178	50
COVERTYPE	1	0.507	50	1	0.400	50	4	0.418	50
IJCNN1	2	0.103	50	2	0.096	50	4	0.095	50
MNIST	1	0.350	50	2	0.193	50	4	0.164	50
HTRU2	1	0.047	50	3	0.032	50	5	0.030	50

TABLE 4.4
Results with three different initial sample size N_0.

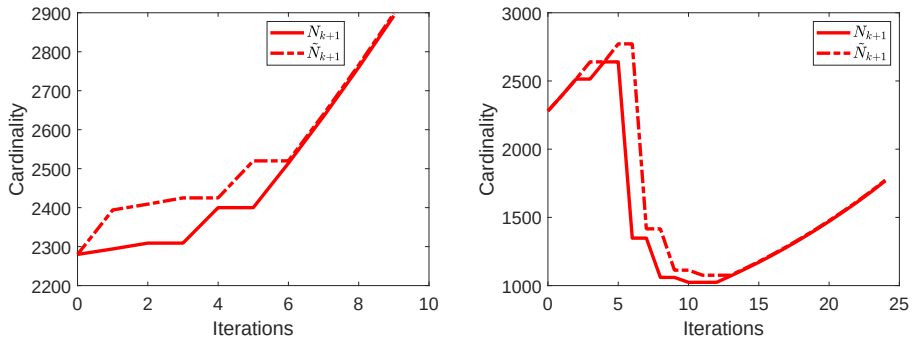

FIG. 4.1. *Dataset* A9A. *Samples sizes N_{k+1} and $\widetilde{N}_{k+1}$ versus iterations with $\mu = 100/N$ (left) and $\mu = 1$ (right), respectively, obtained with a single run of SIRTR. Classification errors:* `err` $=$ *0.187 with $\mu = 100/N$,* `err` $=$ *0.174 with $\mu = 1$.*

reducing N_0, the number of full function/gradient evaluations reduces as well, and that for $N_0 = \lceil 0.01N \rceil$ the average classification error compares well with the error when $N_0 = \lceil 0.1N \rceil$; for instance, the best results for CINA0 and COVERTYPE are obtained by shrinking N_0 to 1% of the maximum sample size. However, choosing $N_0 = \lceil 0.001N \rceil$ affects the classification error which increases of more than 10% in certain cases (see e.g. COVERTYPE or MNIST). The loss of accuracy for some problem may be ascribed to a premature stooping in (4.6) with functions and/or gradients which are not accurate enough. In fact, the stopping criterion (4.6) is the more reliable the more accurate functions and gradient approximations are. We conclude pointing out that again, all runs except one for CINA0 dataset are performed with random function evaluations.

Finally, in Figures 4.1-4.2, we report the plots of the sample sizes N_{k+1}^t and $\widetilde{N}_{k+1}$ with respect to the number of iterations, obtained by running SIRTR on the A9A and MNIST datasets, respectively. In particular, we let either $\mu = 100/N$ or $\mu = 1$ in the update rule (4.3), $\tilde{c} = 1.05$ in (4.2), $c = 0.1$ in (4.4) and $N_0 = \lceil 0.1N \rceil$. Note that a larger μ allows for the decreasing of both N_{k+1}^t and $\widetilde{N}_{k+1}$ in the first iterations, whereas a linear growth rate is imposed only in later iterations. This behaviour is due to the update condition (4.3), which naturally forces N_{k+1}^t to coincide with $\widetilde{N}_{k+1}$ when Δ_k is sufficiently small. For both choices of μ, we see that N_{k+1}^t can grow slower than $\widetilde{N}_{k+1}$ at some iterations, thus reducing the computational cost per iteration of SIRTR.

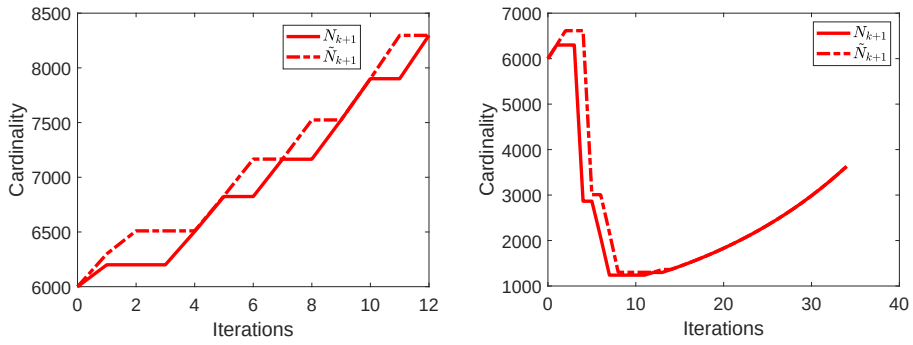

FIG. 4.2. *Dataset* MNIST. *Samples sizes* N_{k+1} *and* $\widetilde{N}_{k+1}$ *versus iterations with* $\mu = 100/N$ *(left) and* $\mu = 1$ *(right), respectively, obtained with a single run of SIRTR. Classification errors:* `err` $= 0.154$ *with* $\mu = 100/N$, `err` $= 0.167$ *with* $\mu = 1$.

4.2. Comparison with TRish. In this section we compare the performance of SIRTR with the so-called Trust-Region-ish algorithm (TRish) recently proposed in [16]. TRish is a stochastic gradient method based on a trust-region methodology. Normalized steps are used in a dynamic manner whenever the norm of the stochastic gradient is within a prefixed interval. In particular, the $k-$th iteration of TRish is given by

$$x_{k+1} = x_k - \begin{cases} \gamma_{1,k}\alpha_k g_k, & \text{if } \|g_k\| \in \left[0, \frac{1}{\gamma_{1,k}}\right) \\ \alpha_k \frac{g_k}{\|g_k\|}, & \text{if } \|g_k\| \in \left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right] \\ \gamma_{2,k}\alpha_k g_k, & \text{if } \|g_k\| \in \left(\frac{1}{\gamma_{2,k}}, \infty\right) \end{cases}$$

where $\alpha_k > 0$ is the steplength parameter, $0 < \gamma_{2,k} < \gamma_{1,k}$ are positive constants, and $g_k \in \mathbb{R}^n$ is a stochastic gradient estimate. This algorithm has proven to be particularly effective on binary classification and neural network training, especially if compared with the standard stochastic gradient algorithm [16, Section 4].

For our numerical tests, we implement TRish with subsampled gradients $g_k = \nabla f_S(x_k)$ defined in (2.4). The steplength is constant, $\alpha_k = \alpha$, $\forall k \geq 0$, and α is chosen in the set $\{10^{-3}, 10^{-1}, \sqrt{10^{-1}}, 1, \sqrt{10}\}$. Following the procedure in [16, Section 4], we use constant parameters $\gamma_{1,k} \equiv \gamma_1$, $\gamma_{2,k} \equiv \gamma_2$ and select γ_1, γ_2 as follows. First, Stochastic Gradient algorithm [26] is run with constant steplength equal to 1; second, the average norm G of stochastic gradient estimates throughout the runs is computed; third γ_1, γ_2 are set as $\gamma_1 = \frac{4}{G}$, $\gamma_2 = \frac{1}{2G}$.

We compare TRish with SIRTR where, based on the previous section, we set $N_0 = \lceil 0.01N \rceil$, $N_{k+1,g} = \lceil 0.1N_{k+1}^t \rceil$, $\widetilde{N}_{k+1} = \min\{N, \lceil 1.05N_k \rceil\}$. In TRish, the sample size S of stochastic gradient estimates is $\lceil 10^{-3}N \rceil$ which corresponds to the first sample size used in SIRTR. We ran each algorithm for ten epochs on the datasets A9A and HTRU2 using the null initial guess. We performed 10 runs to report results on average.

After tuning, the parameter setting for TRish was $\gamma_1 \approx 34.5375$, $\gamma_2 \approx 4.3172$ for A9A, and $\gamma_1 \approx 56.5305$, $\gamma_2 \approx 7.0663$ for HTRU2. In Figures 4.3-4.4, we report the decrease of the (average) classification error, training loss f_N and testing loss, $f_{N_T}(x) = \frac{1}{N_T}\sum_{i \in I_{N_T}} \phi_i(x)$, over the (average) number of full function and gradient

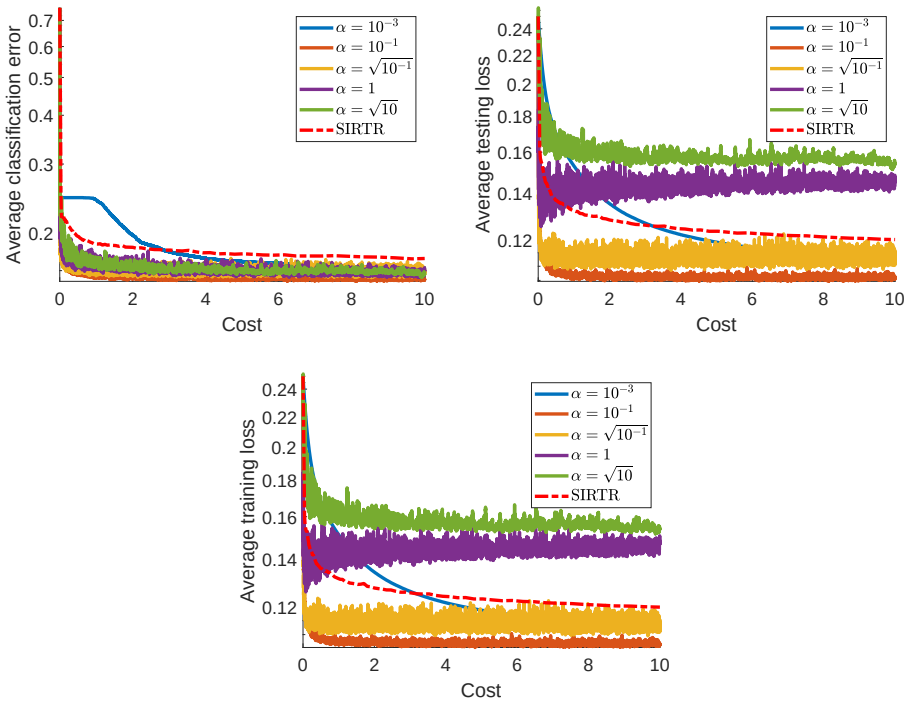

FIG. 4.3. *Dataset* A9A. *Average classification error, testing loss (top row) and training loss (bottom row) versus epochs.*

evaluations required by the algorithms. From these plots, we can see that SIRTR performs comparably to the best implementation of TRish on HTRU2, while showing a good, though not optimal, performance on A9A.

In accordance to the experience in [16], all parameters γ_1 and γ_2 and α are problem-dependent; note that the best performance of TRish is obtained with $\alpha = 10^{-1}$ for A9A and with $\alpha = 10^{-3}$ for HTRU2, respectively. By contrast, SIRTR performs similarly to TRish without requiring parameter tuning, which is the key feature of adaptive stochastic optimization methods.

5. Conclusions. We proposed a stochastic gradient method coupled with a trust-region strategy and an inexact restoration approach for solving finite-sum minimization problems. Functions and gradients are subsampled and the batch size is governed deterministically by the inexact restoration approach and the trust-region acceptance rule until, eventually, full function evaluations are required. We showed the theoretical properties of the method and gave a worst-case complexity result on the number of iterations required to reach full precision in function evaluations. Numerical experience shows that the proposed method provides good results without reaching full precision and thus keeping the overall computational cost relatively low.

REFERENCES

[1] A. S Bandeira, K. Scheinberg, L. N. Vicente, *Convergence of trust-region methods based on probabilistic models*, SIAM Journal on Optimization, 24(3), 1238–1264, 2014.

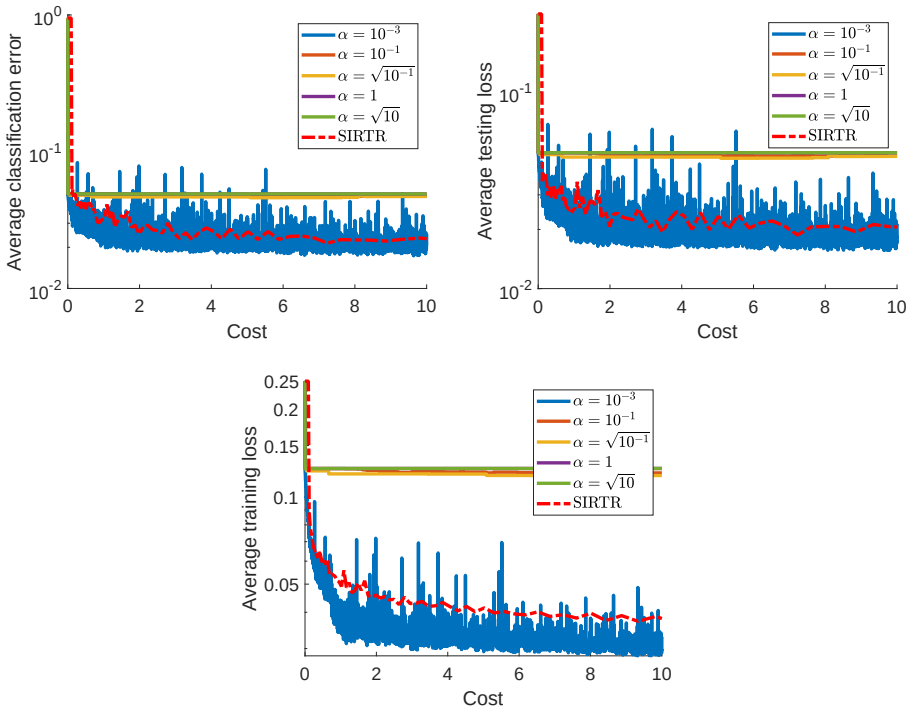

FIG. 4.4. *Dataset* HTRU2. *Average classification error, testing loss (top row) and training loss (bottom row) versus epochs.*

[2] S. Bellavia, T. Bianconcini, N. Krejić, B. Morini, *Subsampled first-order optimization methods with applications in imaging*. Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging. Springer, 1–35, 2021.

[3] S. Bellavia, N. Krejić, B. Morini, *Inexact restoration with subsampled trust-region methods for finite-sum minimization*, Computational Optimization and Applications 76, 701–736, 2020.

[4] S. Bellavia, G. Gurioli, B. Morini, *Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization*, IMA Journal of Numerical Analysis, IMA Journal of Numerical Analysis, 41(1), 764-799, 2021.

[5] S. Bellavia, G. Gurioli, B. Morini, Ph. L. Toint, *Adaptive Regularization Algorithms with Inexact Evaluations for Nonconvex Optimization*, SIAM Journal on Optimization, 29, 2881–2915, 2019.

[6] D. P. Bertsekas, *Nonlinear Programming, 3rd Edition*, Athena Scientific, 2016.

[7] G. E. Birgin, N. Krejić, J. M. Martínez, *On the employment of Inexact Restoration for the minimization of functions whose evaluation is subject to programming errors*, Mathematics of Computation 87(311), 1307–1326, 2018.

[8] G. E. Birgin, N. Krejić, J. M. Martínez, *Iteration and evaluation complexity on the minimization of functions whose computation is intrinsically inexact*, Mathematics of Computation, 89, 253–278, 2020.

[9] J. Blanchet, C. Cartis, M. Menickelly, K. Scheinberg, *Convergence Rate Analysis of a Stochastic Trust Region Method via Submartingales*, INFORMS Journal on Optimization, 1, 92–119, 2019.

[10] L. Bottou, F. C. Curtis, J. Nocedal, *Optimization Methods for Large-Scale Machine Learning,* SIAM Review, 60(2), 223–311, 2018.

[11] R. Bollapragada, R. Byrd, and J. Nocedal, *Adaptive sampling strategies for stochastic optimization*, SIAM Journal on Optimization, 28, 33123343, 2018.

[12] R. H. Byrd, G. M. Chin, J. Nocedal, Y. Wu, *Sample size selection in optimization methods for machine learning*, Mathematical Programming, 134, 127–155, 2012.

[13] V. K. Chauhan, A. Sharma, K. Dahiya, *Stochastic trust region inexact Newton method for large-scale machine learning*, International Journal of Machine Learning and Cybernetics **11**(7), 1541–1555, 2020.

[14] R. Chen, M. Menickelly, K. Scheinberg, *Stochastic optimization using a trust-region method and random models*, Mathematical Programming, 169(2), 447–487, 2018.

[15] F. E. Curtis, K. Scheinberg, *Adaptive Stochastic Optimization: A Framework for Analyzing Stochastic Optimization Algorithms*, IEEE Signal Processing Magazine, 37(5), 32–42, 2020.

[16] F. E. Curtis, K. Scheinberg, R. Shi, *A Stochastic Trust Region Algorithm Based on Careful Step Normalization*, INFORMS Journal on Optimization 1(3), 200–220, 2019.

[17] F. E. Curtis, K. Scheinberg, *Optimization methods for supervised machine learning: From linear models to deep learning*, Leading Developments from INFORMS Communities. INFORMS, 2017. 89–114.

[18] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT Press, http://www.deeplearningbook.org, 2016.

[19] R. M. Gower, M. Schmidt, F. Bach, P. Richtarik, *Variance-reduced methods for machine learning*. Proceedings of the IEEE, 108(11), 1968–1983, 2020.

[20] R. Johnson, T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, Proceedings of the 26th International Conference on Neural Information Processing Systems 26, (NIPS 2013).

[21] D. P. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization*, Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.

[22] N. Krejić, J. M. Martínez, *Inexact Restoration approach for minimization with inexact evaluation of the objective function*, Mathematics of Computation, 85, 1775-1791, 2016.

[23] J. M. Martínez and E. A. Pilotta, *Inexact restoration algorithms for constrained optimization*, Journal of Optimization Theory and Applications, 104, 135–163, 2000.

[24] L. M. Nguyen, J. Liu, K. Scheinberg and M. Takač, *SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient*, Proceedings of the 34th International Conference on Machine Learning, PMLR 70:2613-2621, 2017.

[25] C. Paquette, K. Scheinberg, *A Stochastic Line Search Method with Expected Complexity Analysis*, SIAM J. Optim., 30 349–376, 2020.

[26] H. Robbins, S. Monro, *A Stochastic Approximation Method*, The Annals of Mathematical Statistics, 22 400–407, 1951.

[27] M. Schmidt, N. Le Roux, F. Bach, *Minimizing Finite Sums with the Stochastic Average Gradient*, Math. Program. 162, 83–112, 2017.

[28] W. Xiaoyu, Y. X Yuan, *Stochastic Trust Region Methods with Trust Region Radius Depending on Probabilistic Models*, arXiv:1904.03342, 2019.

[29] P. Xu, F. Roosta-Khorasani, M. W. Mahoney, *Second-order optimization for non-convex machine learning: an empirical study*, Proceedings of the 2020 SIAM International Conference on Data Mining.