Check for updates

A stochastic first-order trust-region method with inexact restoration for finite-sum minimization

Stefania Bellavia[1] · Nataša Krejić[2] · Benedetta Morini[1] · Simone Rebegoldi[1]

Abstract

We propose a stochastic first-order trust-region method with inexact function and gradient evaluations for solving finite-sum minimization problems. Using a suitable reformulation of the given problem, our method combines the inexact restoration approach for constrained optimization with the trust-region procedure and random models. Differently from other recent stochastic trust-region schemes, our proposed algorithm improves feasibility and optimality in a modular way. We provide the expected number of iterations for reaching a near-stationary point by imposing some probability accuracy requirements on random functions and gradients which are, in general, less stringent than the corresponding ones in literature. We validate the proposed algorithm on some nonconvex optimization problems arising in binary classification and regression, showing that it performs well in terms of cost and accuracy, and allows to reduce the burdensome tuning of the hyper-parameters involved.

Keywords Finite-sum minimization · Inexact restoration · Trust-region methods · Subsampling · Worst-case iteration complexity

---

Stefania Bellavia, Nataša Krejić, Benedetta Morini and Simone Rebegoldi have contributed equally to this work.

✉ Benedetta Morini
benedetta.morini@unifi.it

Stefania Bellavia
stefania.bellavia@unifi.it

Nataša Krejić
natasak@uns.ac.rs

Simone Rebegoldi
simone.rebegoldi@unifi.it

1   Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale G.B. Morgagni 40, 50134 Florence, Italy

2   Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovića 4, Novi Sad 21000, Serbia

1 Introduction

In this paper we consider the finite-sum minimization problem

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

where N is very large and finite and $\phi_i : \mathbb{R}^n \to \mathbb{R}$, $1 \le i \le N$, 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. In recent years the need for efficient methods for solving (1) resulted in a large body of literature and a number of methods have been proposed and analyzed, see e.g., the reviews [1–3].

It is common to 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 [4] and more contemporary variants like SVRG [5, 6], SAG [7], ADAM [8] and SARAH [9] are widely used for their simplicity and low cost per-iteration. They do not call for function evaluations but require tuning the learning rate and further possible hyper-parameters such as the mini-batch size. Since the tuning effort may be very computationally demanding [10], more sophisticated approaches use stochastic linesearch or trust-region strategies to adaptively choose the learning rate, see [1, 10–15]. 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) 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 [16] or the trust-region strategy [17–19]; in these papers, function and gradient estimates are built with gradually increasing accuracy and averaging on the same sample.

We propose a novel trust-region method with random models based on the IR methodology. In our proposed method, feasibility and optimality are improved in a modular way, and the resulting procedure differs from the existing stochastic trust-region schemes [13, 14, 20–22] in the acceptance rule for the step. We provide a theoretical analysis and give a bound on the expected iteration complexity to satisfy an approximate first-order optimality condition; this calls for accuracy conditions on random gradients that are assumed to hold with some sufficiently large but fixed probability and are, in general, less stringent than the corresponding ones in [13, 14, 20–22]. Our theoretical analysis improves over the one for the stochastic trust-region method with inexact restoration given in [19], since we no longer rely on standard theory for deterministic unconstrained optimization invoked eventually when functions and gradients are computed exactly.

The paper is organized as follows. In Sect. 2 we give an overview of random models employed in the trust-region framework and introduce the main features of our contribution. The new algorithm is proposed in Sect. 3 and studied theoretically with respect to the iteration complexity analysis. Extensive numerical results are presented in Sect. 4.

2 Trust-region method with random models

Variants of the standard trust-region method based on the use of random models have been presented, to our knowledge, in [13, 14, 19–23]. 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 [20] or replaced by stochastic estimates [13, 14, 19, 21–23].

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. The accuracy requests in the mentioned works show many similarities; here we illustrate some issues related to the works [13, 14, 22], which are closer to our approach.

Let $\|\cdot\|$ denote the 2-norm throughout the paper. At iteration k of a first-order stochastic trust-region model, given x_k, the positive trust-region radius δ_k and a random approximation g_k of $\nabla f_N(x_k)$, let consider the model

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

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 $\min_{\|p\| \leq \delta_k} \varsigma_k(x_k + p)$. Thus, the trust region step takes the form $p_k = -\delta_k g_k / \|g_k\|$.

Two estimates $f^{k,0}$ and $f^{k,p}$ of f_N at x_k and $x_k + p_k$, respectively, are employed to either accept or reject the trial point $x_k + p_k$. The classical ratio between the actual and predicted reduction is replaced by

$$\rho_k = \frac{f^{k,0} - f^{k,p}}{\varsigma_k(x_k) - \varsigma_k(x_k + p_k)}, \tag{2}$$

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 positive and possibly large η_2. Note that the computation of both the step p_k and the denominator in (2) are independent of $f_N(x_k)$. Furthermore, note that a successful iteration might 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.

The condition $\|g_k\| \geq \eta_2 \delta_k$ is not typical of deterministic trust-region methods for smooth optimization and depends on the fact that δ_k controls the accuracy of function and gradients. Specifically, the models used are required to be sufficiently accurate with some probability. The model ς_k is supposed to be, π_M-probabilistically, a κ_*-fully linear model of f_N on the ball $B(x_k, \delta_k)$, i.e., the requirement

$$|f_N(y) - \varsigma_k(y)| \leq \kappa_* \delta_k^2, \quad \|\nabla f_N(y) - g_k\| \leq \kappa_* \delta_k, \quad y \in B(x_k, \delta_k) \tag{3}$$

with $\kappa_* > 0$, has to be fulfilled at least with probability $\pi_M \in (0, 1)$. Moreover, the estimates $f^{k,0}$ and $f^{k,p}$ are supposed to be π_f-probabilistically ϵ_F-accurate estimates of $f_N(x_k)$ and $f_N(x_k + p_k)$, i.e., the requirement

$$|f^{k,0} - f_N(x_k)| \leq \epsilon_F \delta_k^2, \quad |f^{k,p} - f_N(x_k + p_k)| \leq \epsilon_F \delta_k^2, \tag{4}$$

has to be fulfilled at least with probability $\pi_f \in (0, 1)$. Clearly, if f_N is computed exactly then condition (4) is trivially satisfied.

Convergence analysis in [13, 14, 22] shows that for π_M and π_f sufficiently large it holds $\lim_{k \to \infty} \delta_k = 0$ almost surely. Moreover, if f_N is bounded from below and ∇f_N is Lipschitz continuous, then $\lim_{k \to \infty} \|\nabla f_N(x_k)\| = 0$ almost surely. Interestingly, the accuracy in (3) and (4) increases as the trust region radius gets smaller but the probabilities π_M and π_f are fixed.

For problem (1) it is straightforward to build approximations of f_N and ∇f_N by sample average approximations

$$f_M(x) = \frac{1}{M} \sum_{i \in I_M} \phi_i(x), \qquad \nabla f_S(x) = \frac{1}{S} \sum_{i \in I_S} \nabla \phi_i(x), \tag{5}$$

where I_M and I_S are subsets of $\{1, \ldots, N\}$ of cardinality $|I_M| = M$ and $|I_S| = S$, respectively. The choice of sample size such that (3) and (4) hold in probability is discussed in [14, §5] as follows. Let $\mathbb{E}_\omega[|\phi_\omega(x) - f_N(x)|^2] \leq V_f$, $\mathbb{E}_\omega[|\nabla \phi_\omega(x) - \nabla f_N(x)|^2] \leq V_g$, $\forall x \in \mathbb{R}^n$, with $\mathbb{E}_\omega$ being the expected value with respect to the random index ω employed for sampling, and assume

$$M \geq \frac{V_f}{\epsilon_F^2 (1 - \pi_f) \delta_k^4}, \quad S \geq \frac{V_g}{\kappa_*^2 (1 - \pi_g) \delta_k^2} \text{ and } \max\{M, S\} \leq N. \tag{6}$$

Then $f^{k,0}$ and $f^{k,s}$ built as in (5) with sample size M satisfy (4) with probability p_f, while g_k built as in (5) with sample size S satisfies $\|\nabla f_N(x_k) - g_k\| \leq \kappa_* \delta_k$ with probability p_g. Furthermore, using Taylor expansion and Lipschitz continuity of ∇f_N, it can be proved that (3) is met with probability $\pi_M = \pi_f \pi p_g$; consequently, a κ_*-fully linear model of f_N in $B(x_k, \delta_k)$ is obtained.

In principle, conditions (3), (4) and $\lim_{k \to \infty} \delta_k = 0$ imply that $f^{k,0}$, $f^{k,s}$ and g_k 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.

2.1 Our contribution

We propose a trust-region procedure with random models based on (5) and combine it with the inexact restoration (IR) method for constrained optimization [24]. To this end, we make a simple transformation of (1) into a constrained problem. Specifically, letting I_M be an arbitrary nonempty subset of $\{1, \ldots, N\}$ of cardinality $|I_M|$ equal to M, we reformulate problem (1) as

$$\min_{x \in \mathbb{R}^n} f_M(x) = \frac{1}{M} \sum_{i \in I_M} \phi_i(x), \text{ s.t. } M = N. \tag{7}$$

Using the IR strategy allows to improve feasibility and optimality in a modular way and gives rise to a procedure that differs from the existing trust-region schemes in the following respects. First, at each iteration a *reference* sample size is fixed and used as a guess for the approximation of function values. Second, the acceptance rule for the step is based on the condition $\|g_k\| \geq \eta_2 \delta_k$, for some $\eta_2 > 0$, and a sufficient decrease condition on a merit function that measures both the reduction of the objective function and the improvement in feasibility. Finally, the expected iteration complexity to satisfy an approximate first-order optimality condition is given, provided that, at each iteration k, the gradient estimates satisfy accuracy requirements of order $\mathcal{O}(\delta_k)$; such accuracy requirements implicitly govern function approximations and are, in general, less stringent than the corresponding ones in [13, 14, 20–22], as carefully detailed in Sect. 3.

Our theoretical analysis improves over the analysis carried out in [19] for a similar stochastic trust-region coupled with inexact restoration, since here we do not rely on the occurrence of full precision, $M = N$ in (7), reached eventually and do not apply standard theory for unconstrained optimization. In fact, the expected number of iterations until a prescribed accuracy is reached is provided without invoking full precision.

3 The algorithm

In this section we introduce our new algorithm referred to as SIRTR (Stochastic Inexact Restoration Trust Region).

First, we introduce some issues of IR methods. The level of infeasibility with respect to the constraint $M = N$ in (7) is measured by the following function h.

Assumption 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

$$\underline{h} \leq h(M) \text{ if } 1 \leq M < N, \quad \text{and} \quad h(M) \leq \overline{h} \text{ if } 1 \leq M \leq N. \tag{8}$$

One possible choice is $h(M) = (N - M)/N$, $1 \leq M \leq N$.

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

$$\Psi(x, M, \theta) = \theta f_M(x) + (1 - \theta)h(M), \tag{9}$$

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

Our SIRTR algorithm is a trust-region method that employs first-order random models. At a generic iteration 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

$$m_k(p) = f_{N_{k+1}^t}(x_k) + g_k^T p, \tag{10}$$

where g_k is a random estimator to $\nabla f_N(x_k)$. Then, we consider the trust-region problem

$$\min_{\|p\| \leq \delta_k} m_k(p), \tag{11}$$

whose solution is

$$p_k = -\delta_k \frac{g_k}{\|g_k\|}. \tag{12}$$

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 (9).

The proposed method is sketched in Algorithm 3.1 and its steps are now discussed. 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, the flag iflag. If iflag=succ the previous iteration was successful, i.e., $x_k = x_{k-1} + p_{k-1}$, if iflag=unsucc the previous iteration was unsuccessful, i.e., $x_k = x_{k-1}$.

The scheduling procedure for generating the trial sample size N_{k+1}^t consists of Steps 1 and 2 of SIRTR. At Step 1, we determine a reference sample size $\widetilde{N}_{k+1} \leq N$. If iflag=succ, then the infeasibility measure h is sufficiently decreased as stated in (20). If iflag=unsucc, $\widetilde{N}_{k+1}$ is left unchanged from the previous iteration, i.e., $\widetilde{N}_{k+1} = \widetilde{N}_k$. We remark that (20) trivially implies $\widetilde{N}_{k+1} = N$ if $N_k = N$ and that it holds at each iteration, even when it is not explicitly enforced at Step 1 (see forthcoming Lemma 1). In principle $\widetilde{N}_{k+1}$ could be the trial sample size but we aim at giving more freedom to the sample size selection process. Thus, at Step 2, we choose a trial sample size N_{k+1}^t complying with condition (21). On the one hand, such a condition allows the choice $N_{k+1}^t < \widetilde{N}_{k+1}$ in order to reduce the computational effort; on the other hand, the choice $N_{k+1}^t \geq \widetilde{N}_{k+1}$ is also possible in order to satisfy specific accuracy requirements that will be specified later. When $N_{k+1}^t < \widetilde{N}_{k+1}$, condition (21) rules the largest possible distance between N_{k+1}^t and $\widetilde{N}_{k+1}$ in terms of δ_k; in case $N_{k+1}^t \geq \widetilde{N}_{k+1}$, (21) is trivially satisfied.

At Step 3 we form the linear random model (10) and compute its minimizer within the trust-region. Specifically, we fix the cardinality $N_{k+1,g}$ and choose the set of indices $I_{N_{k+1,g}} \subseteq \{1, \dots, N\}$ of cardinality $N_{k+1,g}$. Then, we compute the estimator g_k of $\nabla f_N(x_k)$ as

$$g_k = \frac{1}{N_{k+1,g}} \sum_{i \in I_{N_{k+1,g}}} \nabla \phi_i(x_k) \tag{13}$$

and the solution p_k in (12) of the trust-region subproblem (11). Further, we compute $m_k(p_k)$ where m_k is defined in (10) and

$$f_{N_{k+1}^t}(x_k) = \frac{1}{N_{k+1}^t} \sum_{i \in I_{N_{k+1}^t}} \phi_i(x_k), \tag{14}$$

with $I_{N_{k+1}^t} \subseteq \{1, \dots, N\}$ being a set of cardinality N_{k+1}^t.

At Step 4 we compute the new penalty term θ_{k+1}. The computation relies on the predicted reduction defined as

$$\text{Pred}_k(\theta) = \theta(f_{N_k}(x_k) - m_k(p_k)) + (1 - \theta)(h(N_k) - h(\widetilde{N}_{k+1})), \tag{15}$$

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

$$\text{Pred}_k(\theta) \geq \eta_1(h(N_k) - h(\widetilde{N}_{k+1})). \tag{16}$$

If (16) is satisfied at $\theta = \theta_k$ 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 2).

Step 5 establishes if the iteration is successful or not. To this end, given a point $\hat{x}$ and $\theta \in (0, 1)$, the actual reduction of Ψ at the point $\hat{x}$ has the form

$$\begin{aligned}
\text{Ared}_k(\hat{x}, \theta) &= \Psi(x_k, N_k, \theta) - \Psi(\hat{x}, N_{k+1}^t, \theta) \\
&= \theta(f_{N_k}(x_k) - f_{N_{k+1}^t}(\hat{x})) + (1 - \theta)(h(N_k) - h(N_{k+1}^t)),
\end{aligned} \tag{17}$$

and the iteration is successful whenever the following two conditions are both satisfied

$$\text{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta_1 \text{Pred}_k(\theta_{k+1}) \tag{18}$$

$$\|g_k\| \geq \eta_2 \delta_k. \tag{19}$$

Otherwise the iteration is declared unsuccessful. If the iteration is successful, we accept the step and the trial sample size, set iflag=succ and possibly increase the trust-region radius through (23); the upper bound $\delta_{\max}$ on the trust region size is imposed in (23). In case of unsuccessful iterations, we reject both the step and the trial sample size, set iflag=unsucc and decrease the trust region size.

Concerning conditions (18) and (19), we observe that the former mimics the classical acceptance criterion of standard trust-region methods while the latter drives δ_k to zero as $\|g_k\|$ tends to zero.

Algorithm 3.1: The Stochastic IRTR algorithm

Given $x_0 \in \mathbb{R}^n$, N_0 integer in $(0, N]$, $\theta_0 \in (0, 1)$, $0 < \delta_0 < \delta_{\max}$, $\gamma > 1$, $r \in (0, 1), \eta_1 \in (0, 1)$, $\mu > 0$, $\eta_2 > 0$.

0. Set $k = 0$, `iflag=succ`.
1. If `iflag=succ`
   Find $\widetilde{N}_{k+1}$ such that $N_k \le \widetilde{N}_{k+1} \le N$ and

$$h(\widetilde{N}_{k+1}) \le rh(N_k), \tag{20}$$

   Else set $\widetilde{N}_{k+1} = \widetilde{N}_k$.
2. If $N_k = N$ set $N_{k+1}^t = N$
   Else find N_{k+1}^t such that

$$h(N_{k+1}^t) - h(\widetilde{N}_{k+1}) \le \mu\delta_k^2. \tag{21}$$

3. Choose $N_{k+1,g}$, $I_{N_{k+1,g}} \subseteq \{1, \ldots, N\}$ s.t. $|I_{N_{k+1,g}}| = N_{k+1,g}$.
   Compute g_k as in (13), and set

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

   Compute $f_{N_{k+1}^t}(x_k)$ as in (14), and $m_k(p_k) = f_{N_{k+1}^t}(x_k) + g_k^T p_k$.
4. Compute the penalty parameter θ_{k+1}

$$\theta_{k+1} = \begin{cases} \theta_k & \text{if } \operatorname{Pred}_k(\theta_k) \ge \eta_1(h(N_k) - h(\widetilde{N}_{k+1})) \\ \dfrac{(1 - \eta_1)(h(N_k) - h(\widetilde{N}_{k+1}))}{m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1})} & \text{otherwise.} \end{cases} \tag{22}$$

5. If $\operatorname{Ared}_k(x_k + p_k, \theta_{k+1}) \ge \eta_1\operatorname{Pred}_k(\theta_{k+1})$ and $\|g_k\| \ge \eta_2\delta_k$
   (*successful iteration*) define

$$x_{k+1} = x_k + p_k$$
$$\delta_{k+1} = \min\{\gamma\delta_k, \delta_{\max}\} \tag{23}$$

   set $N_{k+1} = N_{k+1}^t$, $k = k + 1$, `iflag=succ` and go to Step 1.
   Else (*unsuccessful iteration*) define

$$x_{k+1} = x_k$$
$$\delta_{k+1} = \frac{\delta_k}{\gamma}, \tag{24}$$

   set $N_{k+1} = N_k$, $k = k + 1$, `iflag=unsucc` and go to Step 1.

We conclude the description of Algorithm 3.1 showing that condition (20) holds for all iterations, even when it is not explicitly enforced at Step 1.

Lemma 1 *Let Assumption* 3.1 *holds and* $r \in (0, 1)$ *be the scalar in Algorithm* 3.1. *The sample sizes* $\widetilde{N}_{k+1} \leq N$ *and* $N_k \leq N$ *generated by Algorithm* 3.1 *satisfy*

$$h(\widetilde{N}_{k+1}) \leq rh(N_k), \quad \forall k \geq 0. \tag{25}$$

Proof We observe that, by Assumption 3.1, (25) 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 enforce (25) at Step 1. Now consider a generic iteration $\bar{k} \geq 1$ and suppose that (25) holds for $\bar{k} - 1$. If iteration $\bar{k} - 1$ is successful, then condition (25) is enforced for iteration $\bar{k}$ at Step 1.

If iteration $\bar{k} - 1$ is unsuccessful, then at Step 5 we set $N_{\bar{k}} = N_{\bar{k}-1}$. Successively, at Step 1 of iteration $\bar{k}$ we set $\widetilde{N}_{\bar{k}+1} = \widetilde{N}_{\bar{k}}$. Since (25) holds by induction at iteration $\bar{k} - 1$, we have $h(\widetilde{N}_{\bar{k}}) \leq rh(N_{\bar{k}-1})$, which can be rewritten as $h(\widetilde{N}_{\bar{k}+1}) \leq rh(N_{\bar{k}})$ due to the previous assignments at Step 5 and Step 1. Then condition (25) holds also at iteration $\bar{k}$.   □

3.1 On the sequences $\{\theta_k\}$ and $\{\delta_k\}$

In this section, we analyze the properties of Algorithm 3.1. In particular, we prove that the sequence $\{\theta_k\}$ is non increasing and uniformly bounded from below, and that the trust region radius δ_k tends to 0 as $k \to \infty$. We make the following assumption.

Assumption 2 Functions ϕ_i are continuously differentiable for $i = 1, \ldots, n$. There exists $f_{low} \in \mathbb{R}$ such that

$$f_M(x) \geq f_{low}, \quad 1 \leq M \leq N, \, x \in \mathbb{R}^n.$$

Furthermore, there exist $\Omega \subset \mathbb{R}^n$ and $f_{up} \in \mathbb{R}$ such that

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

and all iterates generated by Algorithm 3.1 belong to Ω.

In the following, we let

$$\kappa_\phi = \max\{|f_{low}|, |f_{up}|\}. \tag{26}$$

Remark 1 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 bound constraints (above and below) on all variables; the logistic loss function coupled again with bound constraints (above and below) on the unknowns [25].

In the analysis that follows we will consider two options for $\hat{x}$ in (17), $\hat{x} = x_k + p_k$ for successful iterations and $\hat{x} = x_k$ for unsuccessful iterations.

Our first result characterizes the sequence $\{\theta_k\}$ of the penalty parameters; the proof follows closely [19, Lemma 2.2].

Lemma 2 *Let Assumptions* 1 *and* 2 *hold. Then the sequence* $\{\theta_k\}$ *is positive, non increasing and bounded from below,* $\theta_{k+1} \geq \underline{\theta} > 0$ *with* $\underline{\theta}$ *independent of k and* (16) *holds with* $\theta = \theta_{k+1}$.

Proof We note that $\theta_0 > 0$ and proceed by induction assuming that θ_k is positive. Due to Lemma 1, for all iterations k we have that $N_k \leq \widetilde{N}_{k+1}$ and that $N_k = \widetilde{N}_{k+1}$ 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 $h(N_k) - h(\widetilde{N}_{k+1}) = 0$, and $N_{k+1}^t = N$ by Step 2. Therefore, we have $\mathrm{Pred}_k(\theta) = \theta \delta_k \|g_k\| > 0$ for any positive θ, and (22) implies $\theta_{k+1} = \theta_k$. Let us now consider the case $N_k < \widetilde{N}_{k+1}$. If inequality $\mathrm{Pred}_k(\theta_k) \geq \eta_1(h(N_k) - h(\widetilde{N}_{k+1}))$ holds then (22) gives $\theta_{k+1} = \theta_k$. Otherwise, we have

$$\theta_k\Big(f_{N_k}(x_k) - m_k(p_k) - (h(N_k) - h(\widetilde{N}_{k+1}))\Big) < (\eta_1 - 1)\Big(h(N_k) - h(\widetilde{N}_{k+1})\Big),$$

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, $\mathrm{Pred}_k(\theta) \geq \eta_1(h(N_k) - h(\widetilde{N}_{k+1}))$ is satisfied if

$$\theta(f_{N_k}(x_k) - m_k(p_k) - (h(N_k) - h(\widetilde{N}_{k+1}))) \geq (\eta_1 - 1)(h(N_k) - h(\widetilde{N}_{k+1})),$$

i.e., if

$$\theta \leq \theta_{k+1} \stackrel{\mathrm{def}}{=} \frac{(1 - \eta_1)(h(N_k) - h(\widetilde{N}_{k+1}))}{m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1})}.$$

Hence θ_{k+1} is the largest value satisfying (16) and $\theta_{k+1} < \theta_k$.

Let us now prove that $\theta_{k+1} \geq \underline{\theta}$. Note that by (25) and (8)

$$h(N_k) - h(\widetilde{N}_{k+1}) \geq (1 - r)h(N_k) \geq (1 - r)\underline{h}. \tag{27}$$

Using (26)

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

and θ_{k+1} in (22) satisfies

$$\theta_{k+1} \geq \underline{\theta} = \frac{(1 - \eta_1)(1 - r)\underline{h}}{2k_\phi + \bar{h}}, \tag{28}$$

which completes the proof. □

In the following, we derive bounds for the actual reduction $\mathrm{Ared}_k(x_{k+1}, \theta_{k+1})$ in case of successful iterations and distinguish the iteration indexes k as below:

$$\mathcal{I}_1 = \{k \geq 0 \text{ s.t. } N_k < \widetilde{N}_{k+1}\}, \tag{29}$$

$$\mathcal{I}_2 = \{k \geq 0 \text{ s.t. } N_k = \widetilde{N}_{k+1}\}. \tag{30}$$

Note that $\mathcal{I}_1, \mathcal{I}_2$ are disjoint and any iteration index k belongs to exactly one of these subsets. Moreover, (25) yields $\widetilde{N}_{k+1} = N_k = N_{k+1}^t = N$ for any $k \in \mathcal{I}_2$.

Lemma 3 *Let Assumptions* 1-2 *hold and suppose that iteration k is successful. If $k \in \mathcal{I}_1$ then*

$$\mathrm{Ared}_k(x_{k+1}, \theta_{k+1}) \geq \frac{\eta_1^2(1 - r)\underline{h}}{\delta_{\max}^2} \delta_k^2. \tag{31}$$

Otherwise,

$$\mathrm{Ared}_k(x_{k+1}, \theta_{k+1}) \geq \eta_1 \eta_2 \underline{\theta} \delta_k^2. \tag{32}$$

Proof Since iteration k is successful, $x_{k+1} = x_k + p_k$ and (18) hold. Suppose $k \in \mathcal{I}_1$. By (18) and (16)

$$\mathrm{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta_1 \mathrm{Pred}_k(\theta_{k+1}) \geq \eta_1^2(h(N_k) - h(\widetilde{N}_{k+1})).$$

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

$$\mathrm{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta_1^2(1 - r)h(N_k).$$

Dividing and multiplying the right-hand side above by δ_k^2, applying the inequalities $\underline{h} \leq h(N_k)$, $\delta_k \leq \delta_{\max}$, we get (31).

Suppose $k \in \mathcal{I}_2$. Then $N_k = \widetilde{N}_{k+1}$ and by the definition of $\mathrm{Pred}_k(\theta_{k+1})$ and Lemma 2, we have

$$\mathrm{Pred}_k(\theta_{k+1}) = \theta_{k+1}(f_N(x_k) - m_k(p_k)) = \theta_{k+1}\delta_k \|g_k\| \geq \underline{\theta}\delta_k \|g_k\|,$$

and therefore (18), (19) and Lemma 2 yield (32). □

Let us now define a Lyapunov type function Φ inspired by the paper [14]. Assumption 1 implies that $h(N_k)$ is bounded from above while Assumption 2 implies that $f_{N_k}(x)$ is bounded from below if $x \in \Omega$. Thus, there exists a constant Σ such that

$$f_{N_k}(x) - h(N_k) + \Sigma \geq 0, \quad x \in \Omega, \quad k \geq 0. \tag{33}$$

Definition 1 Let $v \in (0, 1)$ be a fixed constant. For all $k \geq 0$, we define

$$\phi_k \overset{\text{def}}{=} \Phi(x_k, N_k, \theta_k, \delta_k) = v\big(\Psi(x_k, N_k, \theta_k) + \theta_k\Sigma\big) + (1 - v)\delta_k^2, \tag{34}$$

where Ψ is the merit function given in (9) and Σ is given in (33).

The choice of $v \in (0, 1)$ in the above definition will be specified below. First, note that ϕ_k is bounded below for all $k \geq 0$,

$$\begin{aligned}
\phi_k &\geq v\big(\Psi(x_k, N_k, \theta_k) + \theta_k\Sigma\big) \\
&\geq v\big(\theta_k f_{N_k}(x_k) + (1 - \theta_k)h(N_k) + \theta_k(-f_{N_k}(x_k) + h(N_k))\big) \\
&\geq vh(N_k) \geq 0.
\end{aligned} \tag{35}$$

Second, adding and subtracting suitable terms, by the definition (34) and for all $k \geq 0$, we have

$$\begin{aligned}
\phi_{k+1} - \phi_k &= v\Big(\theta_{k+1}f_{N_{k+1}}(x_{k+1}) + (1 - \theta_{k+1})h(N_{k+1})\Big) \\
&\quad - v\big(\theta_k f_{N_k}(x_k) + (1 - \theta_k)h(N_k)\big) + v(\theta_{k+1} - \theta_k)\Sigma \\
&\quad + (1 - v)(\delta_{k+1}^2 - \delta_k^2) \\
&= v\Big(\theta_{k+1}f_{N_{k+1}}(x_{k+1}) + (1 - \theta_{k+1})h(N_{k+1})\Big) \pm v\theta_{k+1}f_{N_k}(x_k) \\
&\quad \pm v(1 - \theta_{k+1})h(N_k) \\
&\quad - v\big(\theta_k f_{N_k}(x_k) + (1 - \theta_k)h(N_k)\big) + v(\theta_{k+1} - \theta_k)\Sigma \\
&\quad + (1 - v)(\delta_{k+1}^2 - \delta_k^2) \\
&= v\Big(\theta_{k+1}(f_{N_{k+1}}(x_{k+1}) - f_{N_k}(x_k)) + (1 - \theta_{k+1})(h(N_{k+1}) - h(N_k))\Big) \\
&\quad + v(\theta_{k+1} - \theta_k)(f_{N_k}(x_k) - h(N_k) + \Sigma) + (1 - v)(\delta_{k+1}^2 - \delta_k^2).
\end{aligned} \tag{36}$$

If the iteration k is successful, then using (33), the monotonicity of $\{\theta_k\}_{k \in \mathbb{N}}$ proved in Lemma 2, and the fact that $N_{k+1} = N_{k+1}^t$, the equality (36) yields

$$\phi_{k+1} - \phi_k \leq -v\text{Ared}_k(x_{k+1}, \theta_{k+1}) + (1 - v)(\delta_{k+1}^2 - \delta_k^2). \tag{37}$$

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 (36) is zero. Hence using again (33) and the monotonicity of $\{\theta_k\}_{k \in \mathbb{N}}$, we obtain

$$\phi_{k+1} - \phi_k \leq (1 - v)(\delta_{k+1}^2 - \delta_k^2). \tag{38}$$

Now we provide bounds for the change of Φ along subsequent iterations and again distinguish the two cases $k \in \mathcal{I}_1, \mathcal{I}_2$ stated in (29)-(30).

Lemma 4 *Let Assumptions* 1 *and* 2 *hold.*

(i) *If the iteration k is unsuccessful, then*

$$\phi_{k+1} - \phi_k \leq \chi_1 \delta_k^2, \qquad \chi_1 = (1-v)\frac{1-\gamma^2}{\gamma^2}. \tag{39}$$

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

$$\phi_{k+1} - \phi_k \leq \chi_2 \delta_k^2, \qquad \chi_2 = \left(-v\left(\frac{\eta_1^2(1-r)\underline{h}}{\delta_{\max}^2}\right) + (1-v)(\gamma^2 - 1)\right). \tag{40}$$

If the iteration k is successful and $k \in \mathcal{I}_2$, *then*

$$\phi_{k+1} - \phi_k \leq \chi_3 \delta_k^2, \qquad \chi_3 = \left(-v\eta_1\eta_2\underline{\theta} + (1-v)(\gamma^2 - 1)\right). \tag{41}$$

Proof (i) If iteration k is unsuccessful, the updating rule (24) for δ_{k+1} implies $\delta_{k+1} = \delta_k/\gamma$. Thus, equation (38) directly yields (39).

(ii) If iteration k is successful, the updating rule (23) for δ_{k+1} implies $\delta_{k+1} \leq \gamma\delta_k$. Thus combining (37) with Lemma 3 we obtain (40) and (41). □

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

Theorem 1 *Let Assumptions* 1 *and* 2 *hold. There exists* $\sigma > 0$, *depending on* $v \in (0, 1)$ *in* (34), *such that*

$$\phi_{k+1} - \phi_k \leq -\sigma\delta_k^2, \quad \text{for all } k \geq 0. \tag{42}$$

Proof In case of unsuccessful iterations, (39) provides a sufficient decrease $\phi_{k+1} - \phi_k$ for any value of $v \in (0, 1)$. In case of successful iterations, χ_2 and χ_3 in (40) and (41) are both negative if

$$\max\left\{\frac{(\gamma^2-1)\delta_{\max}^2}{\eta_1^2(1-r)\underline{h} + (\gamma^2-1)\delta_{\max}^2}, \frac{\gamma^2-1}{\eta_1\eta_2\underline{\theta} + \gamma^2 - 1}\right\} < v < 1. \tag{43}$$

Therefore, if v is chosen as above and

$$\sigma = \min\{\chi_1, \chi_2, \chi_3\}, \tag{44}$$

then (39)–(41) imply (42) and the proof is completed. □

Theorem 2 *Let Assumptions* 1 *and* 2 *hold. Then the sequence* $\{\delta_k\}$ *in Algorithm* 3.1 *satisfies*

$$\lim_{k\to\infty} \delta_k = 0.$$

Proof Under the stated conditions Theorem 1 holds and summing up (42) for $j = 0, 1, \ldots, k-1$, we obtain

$$\phi_k - \phi_0 = \sum_{j=0}^{k-1}(\phi_{j+1} - \phi_j) \le -\sigma \sum_{j=0}^{k-1} \delta_j^2.$$

Given that, by (35), ϕ_k is bounded from below for all k, we conclude that $\sum_{j=0}^{\infty} \delta_j^2 < \infty$, and hence $\lim_{j\to\infty} \delta_j = 0$.   □

3.2 Complexity analysis

Algorithm 3.1 generates a random process since the function estimates in (14) and gradient estimates in (13) are random. All random quantities are denoted by capital letters, while the use of small letters is reserved for their realizations. In particular, the iterates X_k, the trust region radius Δ_k, the steps P_k, the function estimates $F_{N_{k+1}^t}(X_k), F_{N_{k+1}^t}(X_k + P_k)$, the gradient estimates $G_k, \nabla F_{N_{k+1}^t}(X_k)$, and the value Φ_k of the function Φ in (34) at iteration k are random variables, while x_k, δ_k, $p_k, f_{N_{k+1}^t}(x_k), f_{N_{k+1}^t}(x_k + p_k), g_k, \nabla f_{N_{k+1}^t}(x_k), \phi_k$ are their realizations.

In this section, our aim is to derive a bound on the expected number of iterations that occur in Algorithm 3.1 to reach a desired accuracy. We show that our algorithm is included into the stochastic framework given in [13, Section 2] and consequently we derive an upper bound on the expected value of the hitting time $\mathcal{K}_\epsilon$ defined below.

Definition 2 Given $\epsilon > 0$, the hitting time $\mathcal{K}_\epsilon$ is the random variable

$$\mathcal{K}_\epsilon = \min\{k \ge 0 : \|\nabla f_N(X_k)\| \le \epsilon\},$$

i.e., $\mathcal{K}_\epsilon$ is the first iteration such that $\|\nabla f_N(X_k)\| \le \epsilon$.

Our analysis relies on the assumption that g_k and $\nabla f_{N_{k+1}^t}(x_k)$ are probabilistically accurate estimators of the true gradient at x_k, in the sense that the events

$$\mathcal{G}_{k,1} = \{\|\nabla f_N(X_k) - G_k\| \le \nu\Delta_k\}, \tag{45}$$

$$\mathcal{G}_{k,2} = \{\|\nabla f_N(X_k) - \nabla f_{N_{k+1}^t}(X_k)\| \le \nu\Delta_k\}, \tag{46}$$

are true at least with conditioned probability $\pi_1 \in (0, 1)$ and $\pi_2 \in (0, 1)$, respectively. Using the same terminology of [26, 27], we say that iteration k is *true* if both $\mathcal{G}_{k,1}$ and $\mathcal{G}_{k,2}$ are true. Furthermore, we introduce the two random variables

$$I_k = \mathbb{1}_{\mathcal{G}_{k,1}}, \quad J_k = \mathbb{1}_{\mathcal{G}_{k,2}}, \tag{47}$$

where $\mathbb{1}_A$ denotes the indicator function of an event A.

Finally, we need the following additional assumptions.

Assumption 3 The gradients $\nabla \phi_i$ are Lipschitz continuous with constant L_i. Let $L = \frac{1}{2} \max_{1 \leq i \leq N} L_i$.

Under Assumptions 2 and 3, the norm of the gradient estimates $\|g_k\|$ is bounded, as shown below.

Lemma 5 *Let Assumptions 2 and 3 hold. Then there exists $g_{\max}$ such that*

$$\|g_k\| \leq g_{\max}, \quad k \geq 0, \tag{48}$$

where $g_{max} = \sqrt{8L\kappa_\phi}$ and κ_ϕ is given in (26).

Proof By Assumption 3, it easily follows that $\nabla f_{N_{k+1,g}}$ is Lipschitz continuous on $\mathbb{R}^n$ with constant $2L$. Then Assumption 2 and the descent lemma for continuously differentiable functions with Lipschitz continuous gradient [28, Proposition A.24] ensure that

$$f_{low} \leq f_{N_{k+1,g}}(y) \leq f_{N_{k+1,g}}(x) + \nabla f_{N_{k+1,g}}(x)^T(y - x) + L\|y - x\|^2, \quad \forall\, x, y \in \mathbb{R}^n.$$

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

$$f_{low} \leq \min_{y \in \mathbb{R}^n} \zeta(y) \equiv f_{N_{k+1,g}}(x) + \nabla f_{N_{k+1,g}}(x)^T(y - x) + L\|y - x\|^2, \quad \forall\, x \in \mathbb{R}^n.$$

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

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

and equivalently $\|\nabla f_{N_{k+1,g}}(x)\|^2 \leq 4L(f_{N_{k+1,g}}(x) - f_{low})$ for all $x \in \mathbb{R}^n$. Using again Assumption 2, we have $f_{N_{k+1,g}}(x) - f_{low} \leq |f_{N_{k+1,g}}(x)| + |f_{low}| \leq 2\kappa_\phi$, and consequently

$$\|\nabla f_{N_{k+1,g}}(x)\|^2 \leq 8L\kappa_\phi, \quad \forall\, x \in \mathbb{R}^n.$$

Setting $x = x_k$ and $g_{max} = \sqrt{8L\kappa_\phi}$ in the previous inequality, we get the result. $\square$

First, we analyze the occurrence of successful iterations and show that the availability of accurate gradients has an impact on the acceptance of the trial steps. The following lemma establishes that if the iteration k is *true* and δ_k is smaller than a certain threshold, then the iteration is successful. The analysis is presented for a single realization of Algorithm 3.1 and specializes for k in the sets $\mathcal{I}_1$, $\mathcal{I}_2$.

Lemma 6 *Let Assumptions 1–3 hold and suppose that iteration k is true.*

(i) *If $k \in \mathcal{I}_1$, then the iteration is successful whenever*

$$\delta_k \leq \min \left\{ \frac{\|g_k\|}{\eta_3}, \frac{\|g_k\|}{\eta_2} \right\}, \tag{49}$$

where $\eta_3 = \frac{\delta_{\max} g_{\max}(\theta_0(2\nu+L)+(1-\underline{\theta})\mu)}{\eta_1(1-\eta_1)(1-r)\underline{h}}$.

(ii) *If $k \in \mathcal{I}_2$, then the iteration is successful whenever*

$$\delta_k \leq \min \left\{ \frac{(1-\eta_1)\|g_k\|}{2\nu+L}, \frac{\|g_k\|}{\eta_2} \right\}. \tag{50}$$

Proof From Assumption 3, it follows that $\nabla f_{N_{k+1}^t}$ is Lipschitz continuous with constant $2L$. Then,

$$\begin{aligned}
|m_k(p_k) - f_{N_{k+1}^t}(x_k + p_k)| &= \left| \int_0^1 \left(g_k \pm \nabla f_{N_{k+1}^t}(x_k) - \nabla f_{N_{k+1}^t}(x_k + \tau p_k) \right)^T p_k d\tau \right| \\
&\leq \int_0^1 \|g_k - \nabla f_{N_{k+1}^t}(x_k)\| \|p_k\| d\tau + \int_0^1 2L\tau \|p_k\|^2 d\tau \\
&\leq \int_0^1 (\|g_k - \nabla f_N(x_k)\| + \|\nabla f_N(x_k) - \nabla f_{N_{k+1}^t}(x_k)\|) \|p_k\| d\tau \\
&\quad + \int_0^1 2L\tau \|p_k\|^2 d\tau
\end{aligned} \tag{51}$$

and, since $\mathcal{G}_{k,1}$ and $\mathcal{G}_{k,1}$ are both true, (45) and (46) yield

$$|m_k(p_k) - f_{N_{k+1}^t}(x_k + p_k)| \leq (2\nu + L)\delta_k^2. \tag{52}$$

Now, let us analyze condition (18) for successful iterations.

(i) If $k \in \mathcal{I}_1$, by (15), (17) and (16) we obtain

$$
\begin{aligned}
\text{Ared}_k(x_k + p_k, \theta_{k+1}) - \eta_1 \text{Pred}_k(\theta_{k+1}) &= (1 - \eta_1)\text{Pred}_k(\theta_{k+1}) \\
&\quad + \text{Ared}_k(\theta_{k+1}) - \text{Pred}_k(\theta_{k+1}) \\
&= (1 - \eta_1)\text{Pred}_k(\theta_{k+1}) \\
&\quad + \theta_{k+1}(m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)) \\
&\quad + (1 - \theta_{k+1})(h(\widetilde{N}_{k+1}) - h(N^t_{k+1})) \\
&\geq \eta_1(1 - \eta_1)(h(N_k) - h(\widetilde{N}_{k+1})) \\
&\quad + \theta_{k+1}(m_k(p_k) - f_{N^t_{k+1}}(x_k + p_k)) \\
&\quad + (1 - \theta_{k+1})(h(\widetilde{N}_{k+1}) - h(N^t_{k+1})).
\end{aligned}
\tag{53}
$$

Using (52), (21) and $\underline{\theta} \leq \theta_{k+1} \leq \theta_0$, we also have

$$
\begin{aligned}
\theta_{k+1}(f_{N^t_{k+1}}(x_k + p_k) - m_k(p_k)) &+ (1 - \theta_{k+1})(h(N^t_{k+1}) - h(\widetilde{N}_{k+1})) \\
&\leq (\theta_0(2\nu + L) + (1 - \underline{\theta})\mu)\delta_k^2.
\end{aligned}
\tag{54}
$$

Note that the combination of (25), (8), (23) and Lemma 5, guarantees that

$$
h(N_k) - h(\widetilde{N}_{k+1}) \geq (1 - r)h(N_k) \geq \frac{(1 - r)\underline{h}\delta_k\|g_k\|}{\delta_{\max}g_{\max}}.
\tag{55}
$$

Then, from (53), (54), and (55), we have

$$
\begin{aligned}
\text{Ared}_k(x_k + p_k, \theta_{k+1}) - \eta_1 \text{Pred}_k(\theta_{k+1}) &\geq \frac{\eta_1(1 - \eta_1)(1 - r)\underline{h}\delta_k\|g_k\|}{\delta_{\max}g_{\max}} \\
&\quad - (\theta_0(2\nu + L) + (1 - \underline{\theta})\mu)\delta_k^2.
\end{aligned}
$$

Combining this result with (19), the proof is complete.

(ii) Using (15), (17), $k \in \mathcal{I}_2$, we have

$$
\begin{aligned}
\text{Ared}_k(x_k + p_k, \theta_{k+1}) - \eta_1 \text{Pred}_k(\theta_{k+1}) &= (1 - \eta_1)\text{Pred}_k(\theta_{k+1}) \\
&\quad + \text{Ared}_k(\theta_{k+1}) - \text{Pred}_k(\theta_{k+1}) \\
&= (1 - \eta_1)\theta_{k+1}\delta_k\|g_k\| \\
&\quad + \theta_{k+1}(m_k(p_k) - f_N(x_k + p_k))
\end{aligned}
$$

Using (52) we get

$$
\begin{aligned}
\text{Ared}_k(x_k + p_k, \theta_{k+1}) - \eta_1 \text{Pred}_k(\theta_{k+1}) &\geq (1 - \eta_1)\theta_{k+1}\delta_k\|g_k\| \\
&\quad - \theta_{k+1}(2\nu + L)\delta_k^2.
\end{aligned}
\tag{56}
$$

Combining the above inequality with (19), we have proved that the iteration is successful whenever (50) holds. □

We can now guarantee that a successful iteration k occurs whenever k is true, the prefixed accuracy ϵ in Definition 2 has not been achieved at k, and δ_k is below

a certain threshold depending on ϵ. Again, the result is stated for a single realization of the algorithm.

Lemma 7 *Let Assumptions* 1–3 *hold. Suppose that* $\|\nabla f_N(x_k)\| > \epsilon$, *for some* $\epsilon > 0$, *the iteration* k *is true, and*

$$\delta_k < \delta^\dagger := \min\left\{ \frac{\epsilon}{2\nu}, \frac{\epsilon}{2\eta_2}, \frac{\epsilon}{2\eta_3}, \frac{\epsilon(1-\eta_1)}{2(2\nu+L)} \right\}. \tag{57}$$

Then, iteration k *is successful.*

Proof By $\|\nabla f_N(x_k)\| > \epsilon$, the occurrence of $\mathcal{G}_{k,1}$ and (57), we have

$$\|g_k - \nabla f_N(x_k)\| \le \nu\delta_k < \frac{\epsilon}{2},$$

and this yields $\|g_k\| \ge \frac{\epsilon}{2}$. Then, Lemma 6 implies that iteration k is successful. $\square$

We now proceed similarly to [13, Section 2] and analyse the random process $\{(\Phi_k, \Delta_k, W_k)\}_{k \in \mathbb{N}}$ generated by Algorithm 3.1, where Φ_k is the random variable whose realization is given in (34) and W_k is the random variable defined as

$$\begin{cases} W_0 = 1 \\ W_{k+1} = 2\left(I_k J_k - \frac{1}{2}\right), \quad k = 0, 1, \dots \end{cases} \tag{58}$$

Clearly, W_k takes values ± 1. We denote with $\mathbb{P}_{k-1}(\cdot)$ and $\mathbb{E}_{k-1}(\cdot)$ the probability and expected value conditioned to the $\sigma-$algebra generated by $F_{N_1^t}(X_0), \dots, F_{N_k^t}(X_{k-1})$, $\nabla F_{N_1^t}(X_0), \dots, \nabla F_{N_k^t}(X_{k-1}), G_0, \dots, G_{k-1}$. Then, we can prove the following result.

Lemma 8 *Let Assumptions* 1–3 *hold,* ν *as in* (43), $\delta^\dagger$ *as in* (57) *and* $\mathcal{K}_\epsilon$ *as in Definition* 2. *Suppose there exists some* $j_{\max} \ge 0$ *such that* $\delta_{\max} = \gamma^{j_{\max}}\delta_0$, *and* $\delta_0 > \delta^\dagger$. *Assume that the estimators* G_k *and* $\nabla f_{N_{k+1}^t}(X_k)$ *are conditionally independent random variables, and the events* $\mathcal{G}_{k,1}, \mathcal{G}_{k,2}$ *occur with sufficiently high probability, i.e.,*

$$\mathbb{P}_{k-1}(\mathcal{G}_{k,1}) = \pi_1, \quad \mathbb{P}_{k-1}(\mathcal{G}_{k,2}) = \pi_2, \quad \text{and} \quad \pi_3 = \pi_1\pi_2 > \frac{1}{2}. \tag{59}$$

Then,

   (i)   *there exists* $\lambda > 0$ *such that* $\Delta_k \le \delta_0 e^{\lambda \cdot j_{\max}}$ *for all* $k \ge 0$;
   (ii) *there exists a constant* $\delta_\epsilon = \delta_0 e^{\lambda \cdot j_\epsilon}$ *for some* $j_\epsilon \le 0$ *such that, for all* $k \ge 0$,

$$\mathbb{1}_{\{\mathcal{K}_\epsilon > k\}}\Delta_{k+1} \ge \mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} \min\{\Delta_k e^{\lambda W_{k+1}}, \delta_\epsilon\}, \tag{60}$$

where W_{k+1} satisfies

$$\mathbb{P}_{k-1}(W_{k+1} = 1) = \pi_3, \quad \mathbb{P}_{k-1}(W_{k+1} = -1) = 1 - \pi_3; \tag{61}$$

(iii)   *there exists a nondecreasing function $\ell : [0, \infty) \to (0, \infty)$ and a constant $\Theta > 0$ such that, for all $k \geq 0$,*

$$\mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} \mathbb{E}_{k-1}[\Phi_{k+1}] \leq \mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} \Phi_k - \mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} \Theta \ell(\Delta_k). \tag{62}$$

Proof The proof parallels that of [13, Lemma 7].

(i) Since $\delta_{\max} = \gamma^{j_{\max}} \delta_0$, we can set $\lambda = \log(\gamma) > 0$, and the thesis follows from Step 5 of Algorithm 3.1.

(ii) Let us set

$$\delta_\epsilon = \frac{\epsilon}{\xi}, \quad \text{where } \xi \geq \max \left\{ 2\nu, 2\eta_2, 2\eta_3, \frac{2(2\nu + L)}{1 - \eta_1} \right\}, \tag{63}$$

and assume that $\delta_\epsilon = \gamma^{j_\epsilon} \delta_0$, for some integer $j_\epsilon \leq 0$; notice that we can always choose ξ sufficiently large so that this is true. As a consequence, $\Delta_k = \gamma^{i_k} \delta_\epsilon$ for some integer i_k.

When $\mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} = 0$, inequality (60) trivially holds. Otherwise, conditioning on $\mathbb{1}_{\{\mathcal{K}_\epsilon > k\}} = 1$, we can prove that

$$\Delta_{k+1} \geq \min\{\delta_\epsilon, \min\{\delta_{\max}, \gamma \Delta_k\} I_k J_k + \gamma^{-1} \Delta_k (1 - I_k J_k)\}. \tag{64}$$

Indeed, for any realization such that $\delta_k > \delta_\epsilon$, we have $\delta_k \geq \gamma \delta_\epsilon$ and because of Step 5, it follows that $\delta_{k+1} \geq \delta_\epsilon$. Now let us consider a realization such that $\delta_k \leq \delta_\epsilon$. Since $\mathcal{K}_\epsilon > k$ and $\delta_\epsilon \leq \delta^\dagger$, if $I_k J_k = 1$ (i.e., k is true), then we can apply Lemma 7 and conclude that k is successful. Hence, by Step 5, we have $\delta_{k+1} = \min\{\delta_{\max}, \gamma \delta_k\}$. If $I_k J_k = 0$, then we cannot guarantee that k is successful; however, again using Step 5, we can write $\delta_{k+1} \geq \gamma^{-1} \delta_k$. Combining these two cases, we get (64). If we observe that $\delta_{\max} = \gamma^{j_{\max}} \delta_0 \geq \gamma^{j_\epsilon} \delta_0 = \delta_\epsilon$, and recall the definition of W_k in (58), then equation (64) easily yields (60). The probabilistic conditions (61) are a consequence of (59).

(iii) The thesis trivially follows from (42) with $\ell(\Delta) = \Delta^2$ and $\Theta = \sigma$.   ☐

The previous lemma shows that the random process $\{(\Phi_k, \Delta_k, W_k)\}_{k \in \mathbb{N}}$ complies with Assumption 2.1 of [13].

Theorem 3 *Under the assumptions of Lemma 8, we have*

$$\mathbb{E}[\mathcal{K}_\epsilon] \leq \frac{\pi_3}{2\pi_3 - 1} \cdot \frac{\phi_0 \xi^2}{\sigma \epsilon^2} + 1. \tag{65}$$

where ξ is chosen as in (63) and σ is given in (44).

Proof The claim follows directly by [13, Theorem 2]. $\square$

Remark 2 The requirement of (45) and (46) to hold in probability is less stringent than the overall conditions (3) and (4). Analogously to the discussion in Sect. 2, if $\mathbb{E}_\omega[|\nabla\phi_\omega(x) - \nabla f_N(x)|^2] \leq V_g$, then Chebyshev inequality guarantees that events (45) and (46) hold in probability when

$$\frac{V_g}{v^2(1-\pi_1)\delta_k^2} \leq N_{k+1,g} \leq N, \quad \frac{V_g}{v^2(1-\pi_2)\delta_k^2} \leq N_{k+1}^t \leq N.$$

Clearly, $\min\{N_{k+1,g}, N_{k+1}^t\} = \mathcal{O}(\delta_k^{-2})$ and in general these sample sizes are expected to growth slower than in (6).

Finally, the complexity theory presented improves on [19] where the iteration complexity before reaching full precision $M = N$ in (7) is estimated, and thereafter existing iteration complexity results for trust-region methods applied to (1) are invoked.

4 Numerical experience

In this section, we evaluate the numerical performance of SIRTR on some non-convex optimization problems arising in binary classification and regression.

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 tests, we equip SIRTR with $\delta_0 = 1$ 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 [19]. 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

$$\widetilde{N}_{k+1} = \min\{N, \lceil \widetilde{c}N_k \rceil\}, \tag{66}$$

where $1 < \widetilde{c} < 2$ is a prefixed constant factor; note that this choice of $\widetilde{N}_{k+1}$ satisfies (21) with $r = (N - (\widetilde{c} - 1))/N$. At Step 2 the function sample size N_{k+1}^t is computed using the rule

$$N_{k+1}^t = \begin{cases} \lceil \widetilde{N}_{k+1} - \mu N\delta_k^2 \rceil, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N\Delta_k^2 \rceil \in [N_0, 0.95N] \\ \widetilde{N}_{k+1}, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N\Delta_k^2 \rceil < N_0 \\ N, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N\Delta_k^2 \rceil > 0.95N. \end{cases} \tag{67}$$

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 (13) and the sample size $N_{k+1,g}$ is fixed as

$$N_{k+1,g} = \lceil cN_{k+1}^t \rceil, \tag{68}$$

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 on nonconvex binary classification problems. 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

$$\min_{x \in \mathbb{R}^n} f_N(x) = \frac{1}{N} \sum_{i=1}^{N} \left(b_i - \frac{1}{1 + e^{-a_i^T x}} \right)^2, \tag{69}$$

where the nonconvex objective function f_N is obtained by composing a least-squares loss with the sigmoid function.

In Table 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}.

We 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 last issue is crucial because it indicates the ability of the inexact restoration approach to solve (69) with random models and to rule sampling and steplength selection.

The average classification error provided by the final iterate, say x_{fin}, is defined as

Table 1 Data sets used

Data set	Training set		Testing set
	N	n	N_T
A8A [29]	15887	123	6809
A9A [29]	22793	123	9768
CINA0	10000	132	6033
COD-RNA [30]	41675	8	17860
COVERTYPE [29]	464810	54	116202
HTRU2 [29]	10000	8	7898
IJCNN1 [30]	49990	22	91701
MNIST [31]	60000	784	10000
PHISHING [30]	7739	68	3316
REAL-SIM [30]	50616	20958	21693
W7A [30]	17284	300	7408
W8A [30]	34824	300	14925

$$\texttt{err} = \frac{1}{N_T} \sum_{i \in I_{N_T}} |b_i - b_i^{pred}|, \tag{70}$$

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 $b_i^{pred} = \max\{\text{sign}(a_i^T x_{\text{fin}}), 0\}$.

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, $1 \leq i \leq N$, is the scalar product $a_i^T x$: once this product is evaluated, it can be reused for computing $\nabla \phi_i$. Nonetheless, following [32, 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). Note that, once ϕ_i is computed, the corresponding gradient $\nabla \phi_i$ requires only $\frac{1}{N}$ backward propagations. Hence, as in our implementation $I_{N_{k+1,g}} \subseteq I_{N'_{k+1}}$, 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 with $x_0 = (0, 0, \ldots, 0)^T$ as initial guess, and stop it when either a maximum of 1000 iterations is reached or a maximum of 500 full function evaluations is performed or the condition

$$|f_{N_k}(x_k) - f_{N_{k-1}}(x_{k-1})| \leq \epsilon |f_{N_{k-1}}(x_{k-1})| + \epsilon, \tag{71}$$

with $\epsilon = 10^{-3}$, holds for a number of consecutive successful iterations such that the computational effort is equal to the effort needed in three iterations with full function and gradient evaluations.

Table 2 Results with three different rules for computing the sample size $N_{k+1,g}$

$N_{k+1,g}$	$\lceil 0.1 N'_{k+1} \rceil$			$\lceil 0.2 N'_{k+1} \rceil$			N'_{k+1}		
	Cost	Err	Sub	Cost	Err	Sub	Cost	Err	Sub
A8A	20	0.170	15	19	0.171	19	22	0.173	29
A9A	20	0.167	12	17	0.169	18	19	0.172	13
CINA0	72	0.146	0	84	0.140	0	116	0.158	1
COD-RNA	44	0.109	0	42	0.106	1	45	0.119	0
COVTYPE	22	0.425	4	19	0.424	8	20	0.435	5
HTRU2	30	0.024	7	25	0.024	13	32	0.024	16
IJCNN1	22	0.087	0	20	0.088	0	20	0.086	0
MNIST2	22	0.154	10	25	0.151	12	29	0.152	18
PHISHING	48	0.105	0	43	0.108	0	48	0.119	0
REAL-SIM	56	0.268	0	56	0.270	0	57	0.294	0
W7A	15	0.079	22	15	0.079	21	16	0.079	34
W8A	13	0.080	25	13	0.080	23	17	0.080	28

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 perform 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 (70) averaged over the 50 runs, `sub` the number of runs where the method is stopped before reaching full accuracy in function evaluations.

In a first set of experiments, we investigate the choice of $N_{k+1,g}$ by varying the factor $c \in (0, 1]$ in (68). In particular, letting $\widetilde{c} = 1.2$ in (66), $\mu = 100/N$ in (67) and $N_0 = \lceil 0.1N \rceil$ as in [19], we test the values $c \in \{0.1, 0.2, 1\}$. The results obtained are reported in Table 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_{k+1}^t is quite convenient from a computationally point of view. By contrast, the choice $N_{k+1,g} = N_{k+1}^t$ leads to the largest computational costs without providing a significant gain in accuracy. Besides the cost per iteration, equal to $\frac{2N_{k+1}^t}{N}$ in this latter case, we observe that full accuracy in function evaluations is reached very often especially for certain datasets, see e.g., CINA0, COD-RNA, COVER-TYPE, IJCNN1, PHISHING, REAL-SIM. Remarkably, the results in Table 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_{k+1}^t. This task can be achieved controlling the growth of $\widetilde{N}_{k+1}$ which affects N_{k+1}^t by means of (67). 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 (66). Average results are reported in Table 3. We can observe that the fastest growth rate for $\widetilde{N}_{k+1}$ is generally more expensive than the other two choices, while the classification

Table 3 Results with three different rules for computing the sample size $\widetilde{N}_{k+1}$

$\widetilde{N}_{k+1}$	$\min\{N, \lceil 1.05N_k \rceil\}$			$\min\{N, \lceil 1.1N_k \rceil\}$			$\min\{N, \lceil 1.2N_k \rceil\}$		
	cost	err	sub	cost	err	sub	cost	err	sub
A8A	27	0.170	49	18	0.170	44	18	0.171	16
A9A	27	0.164	49	18	0.164	38	20	0.168	12
CINA0	35	0.167	44	44	0.163	13	68	0.151	0
COD-RNA	28	0.117	49	38	0.108	17	45	0.102	0
COVTYPE	12	0.396	50	13	0.392	48	20	0.423	7
HTRU2	30	0.022	46	24	0.022	26	25	0.024	11
IJCNN1	21	0.089	50	16	0.086	49	22	0.088	0
MNIST2	19	0.144	50	18	0.144	42	23	0.152	12
PHISHING	28	0.117	50	30	0.110	23	46	0.103	0
REAL-SIM	36	0.254	50	65	0.272	0	57	0.267	0
W7A	26	0.078	50	18	0.078	46	14	0.079	22
W8A	20	0.079	50	14	0.080	46	13	0.080	26

Table 4 Results with three different initial sample sizes N_0

N_0	$\lceil 0.001N \rceil$			$\lceil 0.01N \rceil$			$\lceil 0.1N \rceil$		
	Cost	Err	Sub	Cost	Err	Sub	Cost	Err	Sub
A8A	30	0.182	50	30	0.169	47	28	0.170	50
A9A	27	0.177	50	28	0.165	50	25	0.165	50
CINA0	43	0.111	37	33	0.133	43	34	0.162	44
COD-RNA	4	0.412	50	25	0.194	50	29	0.114	48
COVTYPE	6	0.406	50	8	0.403	50	12	0.406	50
HTRU2	38	0.036	40	35	0.021	43	31	0.021	47
IJCNN1	24	0.095	50	25	0.095	50	19	0.091	50
MNIST2	18	0.185	50	20	0.160	50	21	0.143	50
PHISHING	4	0.410	50	28	0.163	48	29	0.118	50
REAL-SIM	4	0.188	50	5	0.166	50	35	0.254	50
W7A	28	0.077	50	27	0.077	50	25	0.078	50
W8A	23	0.078	50	23	0.079	50	20	0.079	50

Table 5 Average sample size $\bar{N}_{\text{fin}}$ obtained at the final iteration, relative standard deviation s, minimum and maximum sample sizes $N_{\text{fin}}^{\min}, N_{\text{fin}}^{\max}$ observed at the final iteration. Parameters setting: $N_0 = \lceil 0.01N \rceil$, $N_{k+1,g} = \lceil 0.1N_{k+1}^t \rceil$, $\widetilde{N}_{k+1} = \min\{N, \lceil 1.05N_k \rceil\}$, $\mu = 100/N$

	N	N_0	$\bar{N}_{fin}$	s	$N_{fin}^{\min}$	$N_{fin}^{\max}$
A8A	15888	159	10353	0.17	7407	13309
A9A	22793	228	13637	0.22	6718	18730
CINA0	10000	100	7603	0.23	4771	10000
COD-RNA	7739	78	3210	0.74	578	7054
COVTYPE	464810	4649	54762	0.32	33057	100341
HTRU2	10000	100	7902	0.22	3923	10000
IJCNN1	49990	500	26966	0.23	15408	43508
MNIST2	60000	600	22928	0.34	4383	45684
PHISHING	7739	78	3926	0.63	578	7739
REAL-SIM	50617	507	3721	0.034	3604	4174
W7A	17285	173	10334	0.23	5802	14674
W8A	34825	349	17244	0.19	9005	26360

error is similar for all the three choices. Moreover, significantly for $\tilde{c} = 1.05$ most runs stopped before reaching full function accuracy.

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 (66), $\mu = 100/N$ in (67), and $c = 0.1$ in (68). Results are reported in Table 4. We can see that, reducing N_0, the number of full function/gradient evaluations can further reduce in some datasets, 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 most datasets are obtained by shrinking N_0 to 1% of the maximum sample size. We
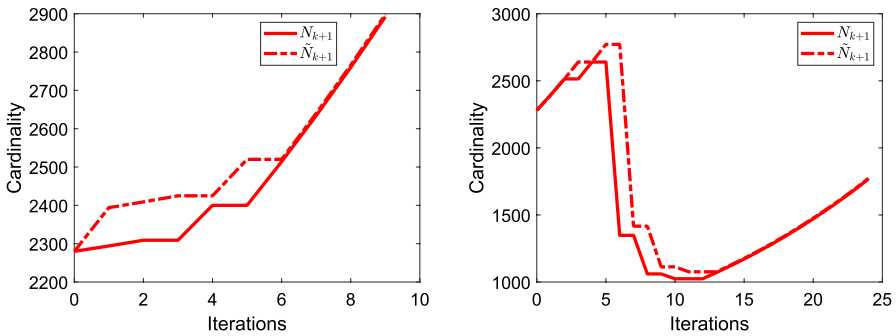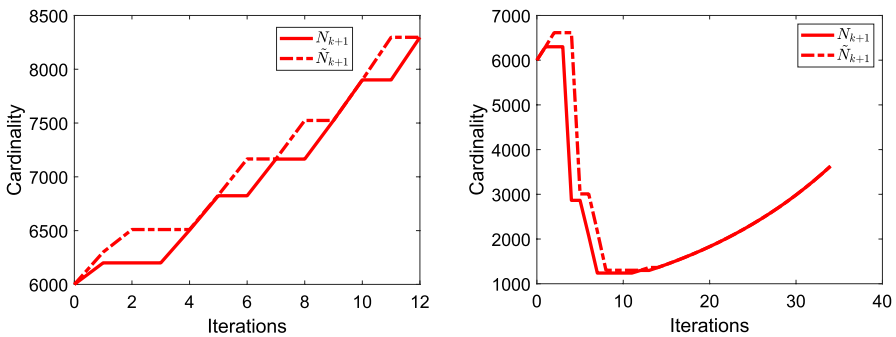
Fig. 1 Dataset A9A. Samples sizes N_{k+1} and $\widetilde{N}_{k+1}$ versus iterations with $\mu = 100/N$ (left) and $\mu = 1$ (right), respectively, obtained with a single run of SIRTR. Classification errors: `err` $= 0.187$ with $\mu = 100/N$, `err` $= 0.174$ with $\mu = 1$

Fig. 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 $\mu = 1$

conclude pointing out that most of the runs are performed without reaching full precision in function evaluation.

As a further confirmation of the efficiency of SIRTR, in Table 5 we report the sample sizes obtained on average at the stopping iteration of SIRTR with parameters setting $N_0 = \lceil 0.01N \rceil$, $N_{k+1,g} = \lceil 0.1N_{k+1}^t \rceil$, $\widetilde{N}_{k+1} = \min\{N, \lceil 1.05N_k \rceil\}$, $\mu = 100/N$. More specifically, for each dataset, we show the mean value $\overline{N}_{\text{fin}}$ obtained by averaging the sample sizes $N_{\text{fin},i}$, $1 \le i \le 50$, used at the final iteration of SIRTR, the relative standard deviation $s = \frac{1}{N_{\text{fin}}} \sqrt{\frac{\sum_{i=1}^{50} (N_{\text{fin},i} - \overline{N}_{\text{fin}})^2}{50}}$ as a measure of dispersion of the final sample sizes with respect to the mean value, and the minimum and maximum sample sizes $N_{\text{fin}}^{\min}, N_{\text{fin}}^{\max}$ observed at the final iteration out of the 50 runs. From the reported values, we deduce that SIRTR terminates with a final sample size which is much smaller, on average, than the maximum sample size N.

Finally, in Figs. 1, 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 (67), $\tilde{c} = 1.05$ in (66), $c = 0.1$ in (68) 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 (67), 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.

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 [33]. 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 [33, Section 4].

For our numerical tests, we implement TRish with subsampled gradients $g_k = \nabla f_S(x_k)$ defined in (5). 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 [33, 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 [4] 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 $\gamma_1 = \frac{4}{G}$, $\gamma_2 = \frac{1}{2G}$.

First, we compare TRish with SIRTR on the nonconvex optimization problem (69), using A9A, HTRU2, MNIST, and PHISHING as datasets (see Table 1). Based on the previous section, we equip SIRTR with $N_0 = \lceil 0.01N \rceil$, $N_{k+1,g} = \lceil 0.1N_{k+1}^t \rceil$, $\widetilde{N}_{k+1} = \min\{N, \lceil 1.05N_k \rceil\}$, $\mu = 100/N$. In TRish, the sample size S of the
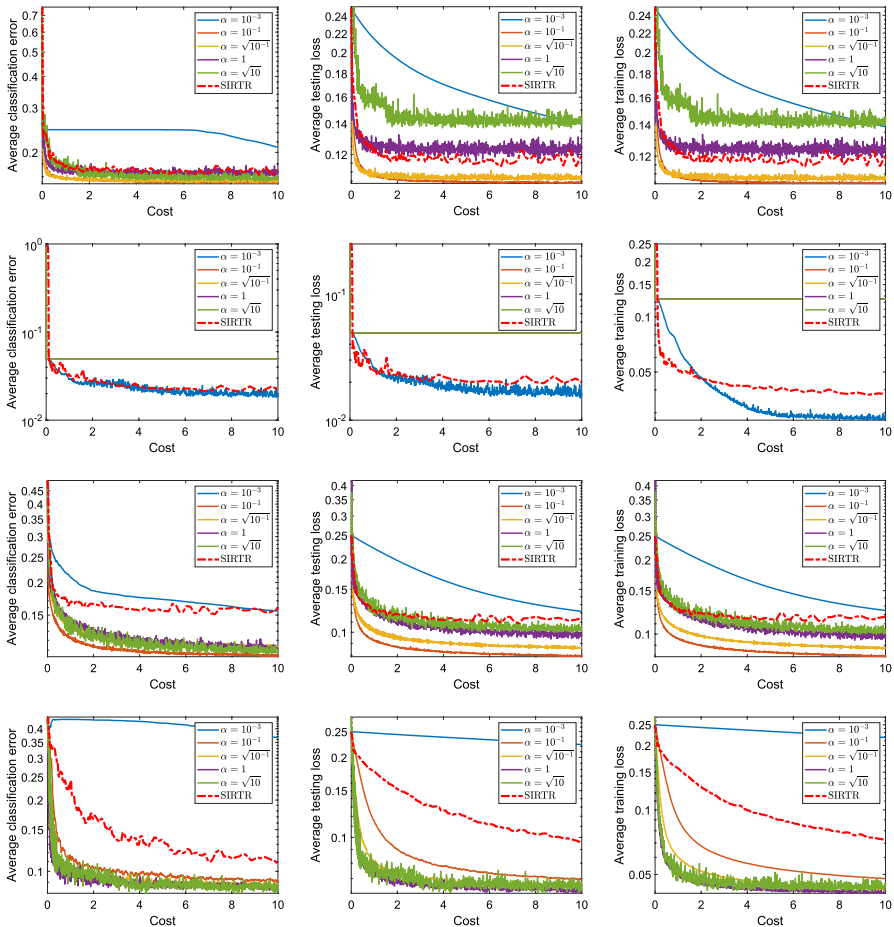
Fig. 3 From top to bottom row: datasets A9A, HTRU2, MNIST, PHISHING. From left to right: Average classification error, testing loss, and training loss versus epochs

stochastic gradient estimates is $\lceil 0.01N \rceil$, which corresponds to the first sample size used in SIRTR. We run each algorithm for ten epochs on the datasets A9A and HTRU2 using the null initial guess. We perform 10 runs to report results on average.

After tuning, the parameter setting for TRish was $\gamma_1 \approx 34.5805$, $\gamma_2 \approx 4.3226$ for A9A, $\gamma_1 \approx 57.9622$, $\gamma_2 \approx 7.2453$ for HTRU2, $\gamma_1 \approx 23.4376$, $\gamma_2 \approx 2.9297$ for MNIST, and $\gamma_1 \approx 50.6409$, $\gamma_2 \approx 6.3301$ for PHISHING. In Fig. 3, 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 evaluations required by the algorithms. From these plots, we can see that SIRTR

performs comparably to the best implementations of TRish on A9A, HTRU2, MNIST, while showing a good, though not optimal, performance on PHISHING.

In accordance to the experience in [33], all parameters γ_1 and γ_2 and α are problem-dependent. For instance, 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 well with an unique setting of the parameters which is the key feature of adaptive stochastic optimization methods.

As a second test, we compare the performance of SIRTR and TRish on a different nonconvex optimization problem arising from nonlinear regression. Letting $\{(a_i, b_i)\}_{i=1}^{N}$ denote the training set, where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ represent the feature vector and the target variable of the i-th example, respectively, we aim at solving the following problem

$$\min_{x \in \mathbb{R}^n} f_N(x) = \frac{1}{N} \sum_{i=1}^{N} \left(b_i - h(a_i; x) \right)^2, \tag{72}$$

where $h(\cdot; x) : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear prediction function.

For this second test, we use the AIR dataset [29], which contains 9358 instances of (hourly averaged) concentrations of polluting gases, as well as temperatures and relative/absolute air humidity levels, recorded at each hour in the period March 2004 - February 2005 from a device located in a polluted area within an Italian city.

As in [34], our goal is to predict the benzene (C6H6) concentration from the knowledge of $n = 7$ features, including carbon monoxide (CO), nitrogen oxides (NO_x), ozone (O_3), non-metanic hydrocarbons (NMHC), nitrogen dioxide (NO_2), air temperature, and relative air humidity. First, we preprocess the dataset by removing examples for which the benzene concentration is missing, reducing the dataset dimension from 9357 to 8991. Then, we employ 70% of the dataset for training ($N = 6294$), and the remaining 30% for testing ($N_T = 2697$). Since the concentration values have been recorded hourly, this means that we use the data measured in the first 9 months for the training phase, and the data related to the
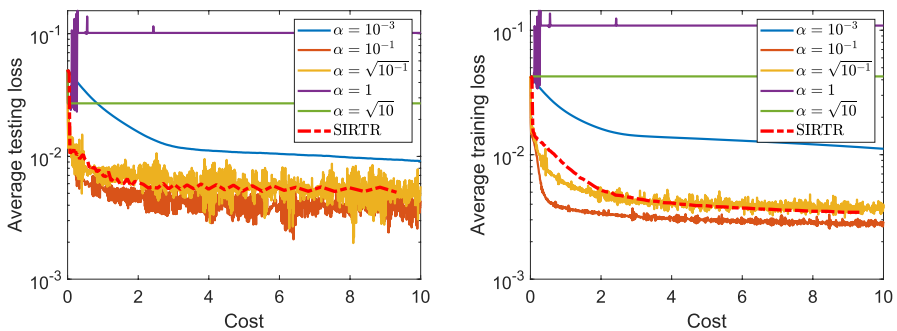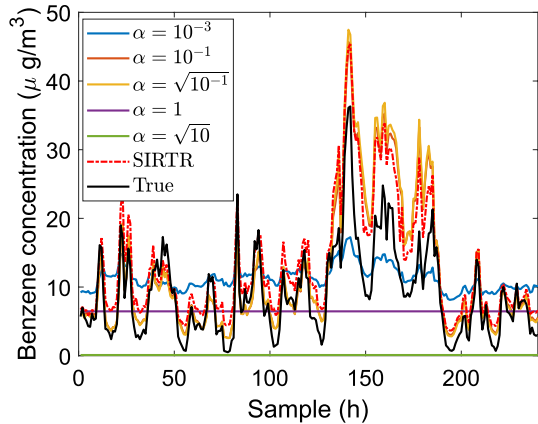

Fig. 4 Dataset AIR. Average testing loss (left) and training loss (right) versus epochs

Fig. 5 Dataset AIR. Estimated concentrations during 10 days (240 hours) compared to the true concentration (black solid line)

last 3 months for the testing phase. Finally, denoting with $D = (d_{ij}) \in \mathbb{R}^{(N+N_T) \times n}$ the matrix containing all the dataset examples along its rows, and setting

$$
\begin{cases}
m_j = \min\limits_{i=1,\dots,N+N_T} d_{ij}, \\
M_j = \max\limits_{i=1,\dots,N+N_T} d_{ij}
\end{cases}, \quad j = 1, \dots, n,
$$

we scale all data values into the interval [0, 1] as follows

$$
d_{ij} = \frac{d_{ij} - m_j}{M_j - m_j}, \quad i = 1, \dots, N + N_T, \, j = 1, \dots, n.
$$

We apply SIRTR and TRish on problem (72), where the prediction function $h(\cdot; x)$ is chosen as a feed-forward neural network based on a $7 \times 5 \times 1$ architecture (see [34] and references therein), with the two hidden layers both equipped with the linear activation function, and the output layer with the sigmoid activation function. We equip the two algorithms with the same parameter values employed in the previous tests, and run them 10 times for 10 epochs, using a random initial guess in the interval $[-\frac{1}{2}, \frac{1}{2}]$.

In Fig. 4, we report the decrease of the (average) training and testing losses provided by SIRTR and by TRish with different choices of the steplength α, whereas in Fig. 5 we show the benzene concentration estimations provided by the algorithms against the true concentration. These results confirm that the performances of SIRTR are comparable with those of TRish equipped with the best choice of the steplength and show the ability of SIRTR to automatically tune the steplength so as to obtain satisfactory results in terms of testing and training accuracy.

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 by the inexact restoration approach and the trust-region acceptance rule. We showed the theoretical properties of the method and gave a worst-case complexity result on the expected number of iterations required to reach an approximate first-order optimality point. Numerical experience showed that the proposed method provides good results keeping the overall computational cost relatively low.

Data availability The dataset CINA0 is no longer available in repositories but is available from the corresponding author on reasonable request. The other datasets analyzed during the current study are available in the repositories: http://www.csie.ntu.edu.tw/~cjlin/libsvm, http://yann.lecun.com/exdb/mnist, https://archive.ics.uci.edu/ml/index.php

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Bellavia, S., Bianconcini, T., Krejić, N., Morini, B.: Subsampled first-order optimization methods with applications in imaging. In: Chen, K., Schonlieb, C., Tai, X., Younces, L. (eds.) Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging. Springer, Switzerland AG (2021)
2. Curtis, F.E., Scheinberg, K.: Optimization methods for supervised machine learning: from linear models to deep learning. Leading Developments from INFORMS Communities. INFORMS, pp. 89–114 (2017)

3. Bottou, L., Curtis, F.C., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018)
4. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
5. Gower, R.M., Schmidt, M., Bach, F., Richtarik, P.: Variance-reduced methods for machine learning. In Proceedings of the IEEE, vol. 108, pp. 1968–1983 (2020)
6. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 1, pp. 315–323 (2013)
7. Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Prog. **162**, 83–112 (2017)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2015)
9. Nguyen, L.M., Liu, J., Scheinberg, K., Takač, M.: Sarah: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In Proceedings of the 34th International Conference on Machine Learning, PMLR, vol. 70, pp. 2613–2621 (2017)
10. Curtis, F.E., Scheinberg, K.: Adaptive stochastic optimization: a framework for analyzing stochastic optimization algorithms. IEEE Signal Process. Mag. **37**(5), 32–42 (2020)
11. Bellavia, S., Gurioli, G., Morini, B.: Adaptive cubic regularization methods with dynamic inexact hessian information and applications to finite-sum minimization. IMA J. Numer. Anal. **41**(1), 764–799 (2021)
12. Bellavia, S., Gurioli, G., Morini, B., Toint, P. L.: Adaptive regularization for nonconvex optimization using inexact function values and randomly perturbed derivatives. J. Complex **68**, 101591 (2022)
13. Blanchet, J., Cartis, C., Menickelly, M., Scheinberg, K.: Convergence rate analysis of a stochastic trust region method via submartingales. INFORMS J. Optim. **1**, 92–119 (2019)
14. Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. Math. Program. **169**(2), 447–487 (2018)
15. Paquette, C., Scheinberg, K.: A stochastic line search method with expected complexity analysis. SIAM J. Optim. **30**, 349–376 (2020)
16. Krejić, N., Martínez, J.M.: Inexact restoration approach for minimization with inexact evaluation of the objective function. Math. Comput. **85**, 1775–1791 (2016)
17. Birgin, G.E., Krejić, N., Martínez, J.M.: On the employment of inexact restoration for the minimization of functions whose evaluation is subject to programming errors. Math. Comput. **87**(311), 1307–1326 (2018)
18. Birgin, G.E., Krejić, N., Martínez, J.M.: Iteration and evaluation complexity on the minimization of functions whose computation is intrinsically inexact. Math. Comput. **89**, 253–278 (2020)
19. Bellavia, S., Krejić, N., Morini, B.: Inexact restoration with subsampled trust-region methods for finite-sum minimization. Comput. Optim. Appl. **76**, 701–736 (2020)
20. Bandeira, A.S., Scheinberg, K., Vicente, L.N.: Convergence of trust-region methods based on probabilistic models. SIAM J. Optim. **24**(3), 1238–1264 (2014)
21. Bellavia, S., Gurioli, G., Morini, B., Toint, P.L.: Trust-region algorithms: probabilistic complexity and intrinsic noise with applications to subsampling techniques. EURO J. Comput. Optim. **10**, 100043 (2022)
22. Xiaoyu, W., Yuan, Y.X.: Stochastic trust region methods with trust region radius depending on probabilistic models. J. Comput. Math. **40**(2), 294–334 (2022)
23. Chauhan, V.K., Sharma, A., Dahiya, K.: Stochastic trust region inexact newton method for large-scale machine learning. Int. J. Mach. Learn. Cybern. **11**(7), 1541–1555 (2020)
24. Martínez, J.M., Pilotta, E.A.: Inexact restoration algorithms for constrained optimization. J. Optim. Theory Appl. **104**, 135–163 (2000)
25. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016) http://www.deeplearningbook.org
26. Beharas, A.S., Cao, L., Scheinberg, K.: Global convergence rate analysis of a generic line search algorithm with noise. SIAM J. Optim. **31**(2), 1489–1518 (2021)
27. Cartis, C., Scheinberg, K.: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Math. Progr. **169**, 337–375 (2018)
28. Bertsekas, D.P. (ed.): Nonlinear Programming. Athena Scientific, Belmont, Massachusetts (2016)
29. Lichman, M.: UCI machine learning repository. (2013) https://archive.ics.uci.edu/ml/index.php

30. Chang, C.C., Lin, C.J.: LIBSVM : a library for support vector machines (2011) http://www.csie.ntu.edu.tw/cjlin/libsvm
31. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In Proceedings of the IEEE, vol. 86, pp. 2278–2324 (1998)
32. Xu, P., Roosta-Khorasani, F., Mahoney, M.W.: Second-order optimization for non-convex machine learning: an empirical study. In Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 199–207 (2020)
33. Curtis, F.E., Scheinberg, K., Shi, R.: A stochastic trust region algorithm based on careful step normalization. INFORMS J. Optim. **1**(3), 200–220 (2019)
34. De Vito, S., Massera, E., Piga, M., Martinotto, L., Di Francia, G.: On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. Sens. Actuator B **129**, 750–757 (2008)