axioms

MDPI

Article

On the Convergence Properties of a Stochastic Trust-Region Method with Inexact Restoration

Stefania Bellavia [1,2,†], Benedetta Morini [1,2,†] and Simone Rebegoldi [1,2,*,†]

[1] Dipartimento di Ingegneria Industriale, Università degli studi di Firenze, Viale G.B. Morgagni 40, 50134 Firenze, Italy
[2] INDAM-GNCS Research Group, P.le Aldo Moro 5, 00185 Roma, Italy
* Correspondence: simone.rebegoldi@unifi.it
† These authors contributed equally to this work.

Abstract: We study the convergence properties of SIRTR, a stochastic inexact restoration trust-region method suited for the minimization of a finite sum of continuously differentiable functions. This method combines the trust-region methodology with random function and gradient estimates formed by subsampling. Unlike other existing schemes, it forces the decrease of a merit function by combining the function approximation with an infeasibility term, the latter of which measures the distance of the current sample size from its maximum value. In a previous work, the expected iteration complexity to satisfy an approximate first-order optimality condition was given. Here, we elaborate on the convergence analysis of SIRTR and prove its convergence in probability under suitable accuracy requirements on random function and gradient estimates. Furthermore, we report the numerical results obtained on some nonconvex classification test problems, discussing the impact of the probabilistic requirements on the selection of the sample sizes.

Keywords: trust-region methods; random models; inexact restoration

MSC: 65K05; 90C30; 90C15

Citation: Bellavia, S.; Morini, B.; Rebegoldi, S. On the Convergence Properties of a Stochastic Trust-Region Method with Inexact Restoration. *Axioms* **2023**, *12*, 38. https://doi.org/10.3390/axioms12010038

Academic Editor: Delfim F. M. Torres

Received: 29 October 2022
Revised: 19 December 2022
Accepted: 24 December 2022
Published: 28 December 2022

1. Introduction

The solutions of large-scale finite-sum optimization problems have become essential in several machine learning tasks including binary or multinomial classification, regression, clustering, and anomaly detection [1,2]. Indeed, the training of models employed in such tasks is often performed by solving the optimization

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

where N is the size of the available data set and the functions $\phi_i : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable for all $i = 1, \ldots, N$. As a result, the efficient solution of a machine learning problem calls for efficient numerical algorithms for (1).

When the data set is extremely large, the evaluation of the objective function f_N and its derivatives may be computationally demanding, making deterministic optimization methods inadequate for solving (1). A common strategy consists of approximating both the function and derivatives by employing a small number of loss functions ϕ_i sampled randomly, making stochastic optimization methods the preferred choice [3–5]. A major issue is the sensitivity of most stochastic algorithms to their parameters, such as the learning rate or sample sizes used for building the function and gradient approximations, which usually need to be tuned through multiple trials and errors before the algorithm exhibits acceptable performance. A possible remedy to burdensome tuning is to employ adaptive optimization methods, which compute the parameters according to appropriate globalization strategies

[6–20]. Most of these methods have probabilistic accuracy requirements for the function and gradient estimates in order to ensure either the iteration complexity of the expectation [7–11,13,15,19,21,22] or the convergence in probability of the iterates [6,12,16–18,20]. In turn, these requirements are reflected in the choice of sample size, which needs to progressively grow as the iterations proceed, resulting in an increasing computational cost per iteration.

In [11], the authors proposed the so-called stochastic inexact restoration trust-region (SIRTR) method for solving (1). SIRTR employs subsampled function and gradient estimates and combines the classical trust-region scheme with the inexact restoration method for constrained optimization problems [23–25]. This combined strategy involves the reformulation of (1) as an optimization problem with two unknown variables x, M, where x is the object to be recovered and M is the sample size of the function estimate, upon which the constraint $M = N$ is imposed. Based on this reformulation of (1), the method acts on the two variables in a modular way: first, it selects the sample size with a deterministic rule aimed at improving *feasibility* with respect to the constraint $M = N$; then, it accepts or rejects the inexact trust-region step by improving *optimality* with respect to a suitable merit function. SIRTR has shown good numerical performance on a series of classification and regression test problems, as its inexact restoration strategy drastically reduces the computational burden due to the selection of the algorithmic parameters. From a theoretical viewpoint, the authors in [11] provided an upper bound on the expected number of iterations to reach a near-stationary point under some appropriate probability accuracy requirements on the random estimators; remarkably, such requirements are less stringent than others employed in the literature. However, the convergence in probability of SIRTR remains unproved, thus leaving open the question of whether the gradient of the objective function in (1) converges to zero with a probability of one. A positive answer to this question would be an important theoretical confirmation of the numerical stability of the method.

In this paper, we improve on the existing theoretical analysis of SIRTR, showing that its iterates drive the gradient to zero with a probability of one. The results will be obtained by combining the theoretical properties of SIRTR with some tools from martingale theory, as typically done for the convergence analysis of adaptive stochastic methods [6,16,18]. Furthermore, we show the numerical results obtained by applying SIRTR on nonconvex binary classification, discussing the impact of the probabilistic accuracy requirements on the performance of the method.

The paper is structured as follows. In Section 2, we briefly outline the method and its main steps. In Section 3, we perform the convergence analysis of the method, showing that its iterates converge with a probability of one. In Section 4, we provide a numerical illustration of a binary classification test problem. Finally, we report the conclusions and future work in Section 5.

Notations: Throughout the paper, $\mathbb{R}$ is the set of real numbers, whereas the symbol $\|\cdot\|$ denotes the standard Euclidean norm on $\mathbb{R}^n$. We denote with $(\Omega, \mathcal{A}, \mathbb{P})$ a probability space, where Ω is the sample space, $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ is the $\sigma-$algebra of events, and $\mathbb{P} : \mathcal{A} \to [0,1]$ is the probability function. Given an event $A \in \mathcal{A}$, the symbol $\mathbb{P}(A)$ stands for the probability of the event A, and $\mathbb{1}_A : \Omega \to \{0,1\}$ denotes the indicator function of an event A, i.e., the function such that $\mathbb{1}_A(\omega) = 1$ if $\omega \in A$, or $\mathbb{1}_A(\omega) = 0$ otherwise. Given a random variable $X : \Omega \to \mathbb{R}$, we denote with $\mathbb{E}(X)$ the expected value of X. Let $X_1, \ldots, X_n$ be n random variables, and the notation $\sigma(X_1, \ldots, X_n)$ stands for the $\sigma-$algebra generated by $X_1, \ldots, X_n$.

2. The SIRTR Method

Here, we present the stochastic inexact restoration trust-region (SIRTR) method, which was formerly proposed in [11]. SIRTR is a trust-region method with subsampled function and gradient estimates, which combines first-order trust-region methodology with the inexact restoration method for constrained optimization [25]. In order to provide a detailed description of SIRTR, we reformulate (1) as the constrained problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f_M(x) = \frac{1}{M}\sum_{i\in I_M}\phi_i(x), \\ &\text{s.t. } M = N \end{aligned} \tag{2}$$

where $I_M \subseteq \{1,\dots,N\}$ is a sample set of the size of the cardinality $|I_M|$ such that $|I_M| = M$. To measure the infeasibility distance of M from the constraint $M = N$, we introduce a function h that measures the distance of $M \in \{1,\dots,N\}$ from N. Function h is supposed to satisfy the following properties.

Assumption 1. *The function $h : \{1,2,\dots,N\} \to \mathbb{R}$ is monotonically decreasing and satisfies $h(1) > 0$, $h(N) = 0$.*

From Assumption 1, the existence of some positive constants $\underline{h}$ and $\overline{h}$ follows such that

$$\underline{h} \le h(M) \le \overline{h}, \quad \text{for } 1 \le M \le N. \tag{3}$$

An example of a function $h : \{1,\dots,N\} \to \mathbb{R}$ satisfying Assumption 1 is $h(M) = \frac{N-M}{N}$.

SIRTR is a stochastic variant of the classical first-order trust-region method, which accepts the trial point according to the decrease of a convex combination of the function estimate f_M with function h. We report SIRTR in Algorithm 1.

At the beginning of each iteration $k \ge 0$, we have at our disposal the iterate x_k, the trust-region radius δ_k, the sample size $N_k \in \{1,\dots,N\}$, the penalty parameter θ_k, and the flag `iflag`, where `iflag = succ` if the previous iteration was successful, in the sense that is specified below, and `iflag = unsucc` otherwise. Then, in Steps 1–5, perform the following tasks.

- In Step 1, if `iflag = succ`, we reduce the current value $h(N_k)$ of the infeasibility measure and find some $\widetilde{N}_{k+1} \in \{1,\dots,N\}$ satisfying $h(\widetilde{N}_{k+1}) \le rh(N_k)$ with $r \in (0,1)$. On the other hand, if `iflag=unsucc`, $\widetilde{N}_{k+1}$ remains the same from one iteration to the other, i.e., we set $\widetilde{N}_{k+1} = \widetilde{N}_k$. Note that $\widetilde{N}_{k+1} = N$ if $N_k = N$.
- Step 2 determines a trial sample size N^t_{k+1} that satisfies $h(N^t_{k+1}) - h(\widetilde{N}_{k+1}) \le \mu\delta_k^2$ with $\mu > 0$ and is used to form the random model. In principle, we could fix $N^t_{k+1} = \widetilde{N}_{k+1}$, but selecting a smaller sample size, if possible, yields a computational saving in the successive step. The relation between N^t_{k+1} and $\widetilde{N}_{k+1}$ depends on δ_k; small values of δ_k give $N^t_{k+1} = \widetilde{N}_{k+1}$; otherwise, N^t_{k+1} is allowed to be smaller than $\widetilde{N}_{k+1}$.
- Step 3 forms the random model $m_k(p)$ and the trial step p_k. The linear model is given by $m_k(p) = f_{N^t_{k+1}}(x_k) + g_k^T p$, where

$$f_{N^t_{k+1}}(x_k) = \frac{1}{N^t_{k+1}}\sum_{i\in I_{N^t_{k+1}}}\phi_i(x_k), \tag{4}$$

and

$$g_k = \frac{1}{N_{k+1,g}}\sum_{i\in I_{N_{k+1,g}}}\nabla\phi_i(x_k), \tag{5}$$

with $I_{N_{k+1,g}} \subset \{1,\dots,N\}$ with cardinality $|I_{N_{k+1,g}}| = N_{k+1,g}$.
Minimizing m_k over the ball of center 0 and radius δ_k gives the trial step

$$p_k = \operatorname*{argmin}_{\|p\|\le\delta_k} m_k(p) = -\delta_k\frac{g_k}{\|g_k\|}.$$

Algorithm 1 Stochastic Inexact Restoration Trust-Region (SIRTR)

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

0. Set $k = 0$, `iflag=succ`.
1. **Reference sample size**
   If `iflag=succ`
   find $\widetilde{N}_{k+1}$ such that $N_k \leq \widetilde{N}_{k+1} \leq N$ and
   $$h(\widetilde{N}_{k+1}) \leq r h(N_k), \tag{6}$$
   Else
   set $\widetilde{N}_{k+1} = \widetilde{N}_k$.
2. **Trial sample size**
   If $N_k = N$
   set $N_{k+1}^t = N$
   Else
   find N_{k+1}^t such that
   $$h(N_{k+1}^t) - h(\widetilde{N}_{k+1}) \quad \leq \quad \mu \delta_k^2. \tag{7}$$
3. **Trust-region model**
   Choose $I_{N_{k+1}^t} \subseteq \{1, \ldots, N\}$ such that $|I_{N_{k+1}^t}| = N_{k+1}^t$.
   Choose $N_{k+1,g}$ and $I_{N_{k+1,g}} \subseteq \{1, \ldots, N\}$ such that $|I_{N_{k+1,g}}| = N_{k+1,g}$.
   Compute g_k as in (5), and set $p_k = -\delta_k \frac{g_k}{\|g_k\|}$.
   Compute $f_{N_{k+1}^t}(x_k)$ as in (4), and set $m_k(p_k) = f_{N_{k+1}^t}(x_k) + g_k^T p_k$.
4. **Penalty parameter**
   If $\mathrm{Pred}_k(\theta_k) \geq \eta_1 (h(N_k) - h(\widetilde{N}_{k+1}))$
   set
   $$\theta_{k+1} = \theta_k$$
   Else
   set
   $$\theta_{k+1} = \frac{(1-\eta_1)(h(N_k) - h(\widetilde{N}_{k+1}))}{m_k(p_k) - f_{N_k}(x_k) + h(N_k) - h(\widetilde{N}_{k+1})}. \tag{8}$$
5. **Acceptance test**
   If $\mathrm{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta_1 \mathrm{Pred}_k(\theta_{k+1})$ and $\|g_k\| \geq \eta_2 \delta_k$ **(success)**
   define
   $$\begin{aligned} x_{k+1} &= x_k + p_k \\ \delta_{k+1} &= \min\{\gamma \delta_k, \delta_{\max}\} \end{aligned} \tag{9}$$
   set $N_{k+1} = N_{k+1}^t$, $k = k+1$, `iflag=succ` and go to Step 1.
   Else **(unsuccess)**
   define
   $$\begin{aligned} x_{k+1} &= x_k \\ \delta_{k+1} &= \frac{\delta_k}{\gamma}, \end{aligned} \tag{10}$$
   set $N_{k+1} = N_k$, $k = k+1$, `iflag=unsucc` and go to Step 1.

- In Step 4, we compute the penalty parameter $\theta_{k+1} \in (0,1)$ that governs the predicted reduction Pred_k in the function and infeasibility measure, which we define as
$$\mathrm{Pred}_k(\theta) = \theta(f_{N_k}(x_k) - m_k(p_k)) + (1-\theta)(h(N_k) - h(\widetilde{N}_{k+1})). \tag{11}$$

If $\theta = \theta_k$ satisfies

$$\text{Pred}_k(\theta) \geq \eta_1(h(N_k) - h(\widetilde{N}_{k+1})), \tag{12}$$

then, we set $\theta_{k+1} = \theta_k$; otherwise, we compute θ_{k+1} as the biggest value for which Inequality (12) is satisfied and takes the explicit form given in (8).

- In Step 5, we establish if we accept (success) or reject (unsuccess) the trial point $x_k + p_k$. The actual reduction Ared_k at point $\hat{x}$ is defined as

$$\text{Ared}_k(\hat{x}, \theta) = \theta(f_{N_k}(x_k) - f_{N_{k+1}^t}(\hat{x})) + (1-\theta)(h(N_k) - h(N_{k+1}^t)), \tag{13}$$

and we declare a successful iteration whenever the following conditions are both met:

$$\text{Ared}_k(x_k + p_k, \theta_{k+1}) \geq \eta_1 \,\text{Pred}_k(\theta_{k+1}) \tag{14}$$

$$\|g_k\| \geq \eta_2 \delta_k. \tag{15}$$

Condition (14) reduces to the standard acceptance criterion of deterministic trust-region methods when $N_k = \widetilde{N}_{k+1} = N_{k+1}^t = N$. If both conditions are satisfied, we accept the step p_k and set $x_{k+1} = x_k + p_k$, increase the trust-region radius based on the update rule (9), and set $N_{k+1} = N_{k+1}^t$, `iflag=succ`; otherwise, we retain the previous iterate, i.e., $x_{k+1} = x_k$, reduce the trust-region radius according to (10), and set $N_{k+1} = N_k$, `iflag = unsucc`.

3. Convergence Analysis

In this section, we are interested in the convergence properties of Algorithm 1. To this aim, we note that the function estimates $f_{N_{k+1}^t}(x_k)$ in (4) and gradient estimates g_k in (5) are all random quantities. Consequently, Algorithm 1 generates a random process, that is, the iterates X_k, the trust region radii Δ_k, the gradient estimates G_k, $\nabla f_{N_{k+1}^t}(X_k)$, and the values Ψ_k of the Lyapunov function Ψ in (21) at iteration k are to be considered as random variables, with their realizations denoted as x_k, δ_k, g_k, and ψ_k.

Our aim is to show the convergence in probability of the iterates generated by Algorithm 1, in the sense that

$$\mathbb{P}\left(\lim_{k\to\infty} \|\nabla f(X_k)\| = 0\right) = 1, \tag{16}$$

i.e., the event $\lim_{k\to\infty} \|\nabla f(X_k)\| = 0$ holds almost surely. We note that the authors in [11] derived a bound on the expected number of iterations in Algorithm 1 required to reach the desired accuracy in the gradient norm, but did not show the convergence results of Type (16).

3.1. Preliminary Results

We recall some technical preliminary results that were obtained for Algorithm 1 in [11]. First, we impose some basic assumptions on the functions in Problem (1).

Assumption 2. *(i) Each function $\phi_i : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable for $i = 1, \ldots, N$.*

(ii) The functions $f_M : \mathbb{R}^n \to \mathbb{R}$, $M = 1, \ldots, N$, are bounded from below on $\mathbb{R}^n$, i.e., there exists $f_{low} \in \mathbb{R}$ such that

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

(iii) The functions $f_M : \mathbb{R}^n \to \mathbb{R}$, $M = 1, \ldots, N$, are bounded from above on a subset $\Omega \subseteq \mathbb{R}^n$, i.e., there exists $f_{up} \in \mathbb{R}$ such that

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

Furthermore, the iterates $\{x_k\}_{k\in\mathbb{N}}$ defined by Algorithm 1 are contained in Ω.

Combining Step 4 of Algorithm 1 with Bound (3) and Assumption 2(iii), it is possible to prove that for any realizations of the algorithm, the sequence $\{\theta_k\}_{k\in\mathbb{N}}$ is bounded away from zero.

Lemma 1 ([11], Lemma 2)**.** *Let Assumptions 1 and 2 hold and consider a particular realization of Algorithm 1. Let $\kappa_\phi > 0$ be defined as follows:*

$$\kappa_\phi = \max\{|f_{low}|, |f_{up}|\}. \tag{17}$$

Then, $\{\theta_k\}_{k\in\mathbb{N}}$ is a positive, non-increasing sequence such that

$$\theta_k \geq \underline{\theta} = \min\left\{\theta_0, \frac{(1-\eta_1)(1-r)\underline{h}}{2k_\phi + \overline{h}}\right\}, \quad \forall\, k \geq 0. \tag{18}$$

Furthermore, Condition (12) holds with $\theta = \theta_{k+1}$.

Since the acceptance test in Algorithm 1 employs function and gradient estimates, we cannot expect that the objective function f_N is decreased from one iteration to the other; however, the authors in [11] showed that an appropriate Lyapunov function Ψ is reduced at each iteration. This Lyapunov function is defined as

$$\Psi(x, M, \theta, \delta) = v(\theta f_M(x) + (1-\theta)h(M) + \theta\Sigma) + (1-v)\delta^2, \tag{19}$$

where $v \in (0,1)$ and $\Sigma \in \mathbb{R}$ are any constants that satisfy

$$f_{N_k}(x) - h(N_k) + \Sigma \geq 0, \quad x \in \Omega, \quad k \geq 0, \tag{20}$$

where such a constant exists thanks to Bound (3) and Assumption 2(ii). For all $k \geq 0$, we denote the values of Ψ along the iterates of Algorithm 1 as follows:

$$\psi_k = \Psi(x_k, N_k, \theta_k, \delta_k), \quad \forall\, k \geq 0. \tag{21}$$

Thanks to (20) and the positive sign of h (see Assumption 1), we can easily deduce that the sequence $\{\psi_k\}_{k\in\mathbb{N}}$ is non-negative, indeed

$$\psi_k \geq v\big(\theta_k f_{N_k}(x_k) + (1-\theta_k)h(N_k) + \theta_k(-f_{N_k}(x_k) + h(N_k))\big) = vh(N_k) \geq 0. \tag{22}$$

Furthermore, the difference between two successive values ψ_{k+1} and ψ_k can be easily rewritten as

$$\begin{aligned}\psi_{k+1} - \psi_k \quad = \quad & v\big(\theta_{k+1}(f_{N_{k+1}}(x_{k+1}) - f_{N_k}(x_k)) + (1-\theta_{k+1})(h(N_{k+1}) - h(N_k))\big) \\ & + v(\theta_{k+1} - \theta_k)(f_{N_k}(x_k) - h(N_k) + \Sigma) + (1-v)(\delta_{k+1}^2 - \delta_k^2).\end{aligned} \tag{23}$$

If k is a successful iteration, then $N_{k+1} = N_{k+1}^t$. By recalling (20) and the fact that the sequence $\{\theta_k\}_{k\in\mathbb{N}}$ is monotone non-increasing (see Lemma 1), then, Equality (23) yields

$$\psi_{k+1} - \psi_k \leq -v\mathrm{Ared}_k(x_{k+1}, \theta_{k+1}) + (1-v)(\delta_{k+1}^2 - \delta_k^2). \tag{24}$$

Otherwise, Algorithm 1 sets $x_{k+1} = x_k$ and $N_{k+1} = N_k$. By inserting these updates in (23), together with (20) and the fact that $\{\theta_k\}_{k\in\mathbb{N}}$ is non-increasing, we obtain

$$\psi_{k+1} - \psi_k \leq (1-v)(\delta_{k+1}^2 - \delta_k^2). \tag{25}$$

Using (24) and (25) in combination with Step 5 of Algorithm 1, we can prove the following results.

Theorem 1 ([11], Theorem 1). *Let Assumptions 1–2 hold and consider a particular realization of Algorithm 1. In (19), choose $v \in (v^\dagger, 1)$, where $v^\dagger$ is defined by*

$$v^\dagger = \max\left\{ \frac{(\gamma^2-1)\delta_{\max}^2}{\eta_1^2(1-r)\underline{h} + (\gamma^2-1)\delta_{\max}^2}, \frac{\gamma^2-1}{\eta_1\eta_2\underline{\theta} + \gamma^2 - 1} \right\}. \tag{26}$$

Then, there exists a constant $\sigma = \sigma(v) > 0$ such that

$$\psi_{k+1} - \psi_k \leq -\sigma\delta_k^2, \quad \text{for all } k \geq 0, \tag{27}$$

hence, the sequence $\{\delta_k\}$ in Algorithm 1 satisfies

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

We now introduce a Lipschitz continuity assumption on the gradients of the functions ϕ_i appearing in (1).

Assumption 3. *Each gradient $\nabla\phi_i$ is L_i-Lipschitz continuous for $i = 1, \ldots, N$. We use the notation $L = \frac{1}{2}\max_{1\leq i\leq N} L_i$.*

The gradient estimates are bounded under Assumptions 2 and 3, as stated in the following lemma.

Lemma 2 ([11], Lemma 5). *Let Assumptions 2 and 3 hold. Then, there exists $g_{\max}$ such that for any realization of Algorithm 1*

$$\|g_k\| \leq g_{\max}, \quad k \geq 0, \tag{28}$$

where $g_{max} = \sqrt{8L\kappa_\phi}$, and κ_ϕ is given in (17).

Let us introduce the following events

$$\mathcal{G}_{k,1} = \{\|\nabla f_N(X_k) - G_k\| \leq \nu\Delta_k\}, \tag{29}$$

$$\mathcal{G}_{k,2} = \{\|\nabla f_N(X_k) - \nabla f_{N_{k+1}^t}(X_k)\| \leq \nu\Delta_k\}, \tag{30}$$

where ν is a positive parameter. By using a similar terminology to the one employed in [22], the iteration k is said to be *true* if the events $\mathcal{G}_{k,1}$ and $\mathcal{G}_{k,2}$ are both true.

The next lemma shows that k is successful whenever the iteration k is true and the trust-region radius δ_k is sufficiently small. This result is crucial for the analysis in the next section.

Lemma 3 ([11], Lemma 6). *Let Assumptions 1–3 hold and set $\eta_3 = \frac{\delta_{\max} g_{\max}(\theta_0(2\nu+L)+(1-\underline{\theta})\mu)}{\eta_1(1-\eta_1)(1-r)\underline{h}}$. Suppose that, for a particular realization of Algorithm 1, the iteration k is true and the following condition holds*

$$\delta_k < \min\left\{ \frac{\|g_k\|}{\eta_2}, \frac{\|g_k\|}{\eta_3}, \frac{(1-\eta_1)\|g_k\|}{2\nu + L} \right\}. \tag{31}$$

Then, iteration k is successful.

3.2. Novel Convergence Results

Here, we derive two novel convergence results in probability holding for Algorithm 1. The results are provided under the assumption that the random variables G_k and $\nabla f_{N_{k+1}^t}(X_k)$ are sufficiently accurate estimators of the true gradient at X_k, in the probabilistic sense specified below.

Assumption 4. *Let* $\mathcal{F}_{k-1} = \sigma(G_0, \ldots, G_{k-1}, \nabla f_{N_1^t}(X_0), \ldots, \nabla f_{N_k^t}(X_{k-1}))$. *Then, the events* $\mathcal{G}_{k,1}, \mathcal{G}_{k,2}$ *are true with sufficiently high probability conditioned to* $\mathcal{F}_{k-1}$, *and the estimators* G_k *and* $\nabla f_{N_{k+1}^t}(X_k)$ *are conditionally independent random variables given* $\mathcal{F}_{k-1}$, *i.e.,*

$$\mathbb{P}(\mathcal{G}_{k,1}|\mathcal{F}_{k-1}) = \pi_1, \quad \mathbb{P}(\mathcal{G}_{k,2}|\mathcal{F}_{k-1}) = \pi_2, \quad \text{and } \pi_3 = \pi_1\pi_2 > \frac{1}{2}. \tag{32}$$

First, we provide a liminf-type convergence result for SIRTR, which shows that the gradient of the objective function converges in probability to zero relative to a subsequence of the iterates.

Theorem 2. *Suppose that Assumptions 1–4 hold. Then, there holds*

$$\mathbb{P}\left(\liminf_{k\to\infty} \|\nabla f(X_k)\| = 0\right) = 1.$$

Proof. The proof parallels that in ([16], Theorem 4.16). By contrast, assume that there exists $\epsilon > 0$ such that the event

$$\|\nabla f(X_k)\| \geq \epsilon, \quad \forall\, k \geq 0 \tag{33}$$

holds with positive probability. Then, let $\{x_k\}_{k\in\mathbb{N}}$ be a realization of $\{X_k\}_{k\in\mathbb{N}}$ such that $\|\nabla f(x_k)\| \geq \epsilon$ for all $k \geq 0$, and $\{\delta_k\}_{k\in\mathbb{N}}$ is the corresponding realization of $\{\Delta_k\}_{k\in\mathbb{N}}$. From Theorem 1, we know that $\lim_{k\to\infty} \delta_k = 0$; therefore, there exists $\bar{k}$ such that

$$\delta_k < b = \min\left\{\frac{\epsilon}{2\nu}, \frac{\epsilon}{2\eta_2}, \frac{\epsilon}{2\eta_3}, \frac{\epsilon(1-\eta_1)}{2(2\nu+L)}, \frac{\delta_{\max}}{\gamma}\right\}, \quad \forall\, k \geq \bar{k}. \tag{34}$$

Consider the random variable R_k with realizations given by

$$r_k = \log_\gamma\left(\frac{\delta_k}{b}\right), \quad k \geq 0. \tag{35}$$

Note that r_k satisfies the following properties.

(i) If $k \geq \bar{k}$, then $r_k \leq 0$; this is a consequence of (34).

(ii) If k is a true iteration and $k \geq \bar{k}$, then $r_{k+1} = r_k + 1$; indeed, since $\mathcal{G}_{k,1}$ is true and $\delta_k < \epsilon/(2\nu)$, it follows that

$$\|g_k - \nabla f_N(x_k)\| \leq \nu\delta_k < \frac{\epsilon}{2}. \tag{36}$$

Then, $\|\nabla f(x_k)\| \geq \epsilon$ yields

$$\|g_k\| \geq \frac{\epsilon}{2}, \tag{37}$$

which, combined with (34), implies that δ_k satisfies Inequality (31). Thus, Lemma 3 implies that the iteration k is successful. Since $\delta_k \leq \delta_{\max}/\gamma$ and the k-th iteration is successful, by (9) it follows that $\delta_{k+1} = \gamma\delta_k$. Hence, $r_{k+1} = r_k + 1$.

(iii) If k is not a true iteration and $k \geq \bar{k}$, then $r_{k+1} \geq r_k - 1$; this is because since we cannot apply Lemma 3 (k is not true), all we can say about the trust-region radius is that $\delta_{k+1} \geq \delta_k/\gamma$.

Then, defining the σ-algebra $\mathcal{F}_{k-1}^{\mathcal{G}} = \sigma(\mathbb{1}_{\mathcal{G}_{0,1}} \cdot \mathbb{1}_{\mathcal{G}_{0,2}}, \ldots, \mathbb{1}_{\mathcal{G}_{k-1,1}} \cdot \mathbb{1}_{\mathcal{G}_{k-1,2}})$, which is included in $\mathcal{F}_{k-1}$, it follows from properties (ii)–(iii) and Assumption 4 that

$$\mathbb{E}(R_{k+1}|\mathcal{F}_{k-1}^{\mathcal{G}}) \geq \pi_1\pi_2(R_k + 1) + (1 - \pi_1\pi_2)(R_k - 1) \geq R_k,$$

where the second inequality follows from $\pi_1\pi_2 > \frac{1}{2}$. Hence, we have that $\{R_k\}_{k\in\mathbb{N}}$ is a submartingale. We also define the random variable

$$W_k = \sum_{i=0}^{k}(2 \cdot \mathbb{1}_{\mathcal{G}_{i,1}} \cdot \mathbb{1}_{\mathcal{G}_{i,2}} - 1), \quad k \geq 0. \tag{38}$$

$\{W_k\}_{k\in\mathbb{N}}$ is also a submartingale as

$$\begin{aligned}\mathbb{E}(W_{k+1}|\mathcal{F}_{k-1}^{\mathcal{G}}) &= \mathbb{E}(W_k|\mathcal{F}_{k-1}^{\mathcal{G}}) + 2\mathbb{E}(\mathbb{1}_{\mathcal{G}_{k+1,1}} \cdot \mathbb{1}_{\mathcal{G}_{k+1,2}}|\mathcal{F}_{k-1}^{\mathcal{G}}) - 1\\ &= W_k + 2\mathbb{P}(\mathcal{G}_{k+1,1} \cap \mathcal{G}_{k+1,2}|\mathcal{F}_{k-1}^{\mathcal{G}}) - 1\\ &\geq W_k,\end{aligned}$$

where, again, the last inequality is due to the fact that $\pi_1\pi_2 > \frac{1}{2}$. Since W_k cannot have a finite limit, from ([16], Theorem 4.4) it follows that the event $\limsup_{k\to\infty} W_k = \infty$ holds almost surely. Since we have $r_k - r_{k_0} \geq w_k - w_{k_0}$ by definition of $\{R_k\}_{k\in\mathbb{N}}$ and $\{W_k\}_{k\in\mathbb{N}}$, it follows that R_k has to be positive infinitely often with a probability of one. However, this contradicts property (i) listed above, which allows us to conclude that (33) cannot occur. □

In the following, we show that SIRTR generates iterates such that the corresponding gradients evaluated at the SIRTR iterates converge (in probability) to zero. The next lemma is similar to ([6], Lemma 4.2); however, some crucial modifications are needed here; indeed, unlike in [6], we take into account the fact that SIRTR enforces the decrease in the Lyapunov function Ψ defined in (19) rather than the objective function.

Lemma 4. *Suppose that Assumptions 1–4 hold. Let $\{X_k\}$ and $\{\Delta_k\}$ be the random sequences generated by Algorithm 1. For a fixed $\epsilon > 0$, define the following subset of natural numbers*

$$\{K_i\} = \{k \geq 0 : \ \|\nabla f(X_k)\| > \epsilon\}.$$

Then,

$$\sum_{k\in\{K_i\}} \Delta_k < \infty.$$

holds almost surely.

Proof. Let us consider the generic realizations $\{x_k\}_{k\in\mathbb{N}}$, $\{g_k\}_{k\in\mathbb{N}}$, $\{\delta_k\}_{k\in\mathbb{N}}$, $\{\theta_k\}_{k\in\mathbb{N}}$, and $\{k_i\}_{i\in\mathbb{N}}$ of Algorithm 1. Furthermore, we let $\{p_i\}$ be the subsequence of $\{k_i\}$, where the iteration is true, whereas $\{n_i\}$ denotes the complementary subsequence so that $\{k_i\} = \{p_i\} \cup \{n_i\}$. First, we show that $\sum_{k\in\{p_i\}} \delta_k < \infty$. If $\{p_i\}$ is finite, then there is nothing to prove. Otherwise, since $\lim_{k\to\infty} \delta_k = 0$, there exists $\widetilde{k}$ such that $\delta_k < b$ for all $k \geq \widetilde{k}$, where b is given in (34). Let us consider any $p_i \geq \widetilde{k}$. Since $\mathcal{G}_{k,1}$ is true, $\delta_{p_i} < \epsilon/(2\nu)$, and $\|\nabla f(x_{p_i})\| > \epsilon$, we can reason as in (36) and (37) to conclude that $\|g_{p_i}\| \geq \epsilon/2$. Combining this lower bound with $\delta_{p_i} < b$, we have that Inequality (31) is satisfied with $k = p_i$. Hence, iteration p_i is successful by Lemma 3 and we have

$$\begin{aligned}\mathrm{Ared}_k(x_{p_i+1}, \theta_{p_i+1}) &\geq \eta_1 \,\mathrm{Pred}_k(\theta_{p_i+1})\\ &\geq \eta_1^2(h(N_{p_i}) - h(\widetilde{N}_{p_i+1}))\\ &\geq \eta_1^2(1-r)h(N_{p_i})\\ &\geq \frac{\eta_1^2(1-r)\underline{h}}{g_{\max}\delta_{\max}}\delta_{p_i}\|g_{p_i}\|,\end{aligned} \tag{39}$$

where the first inequality is the acceptance test (14), the second follows from Step 4 of the SIRTR algorithm, the third follows from (6), and the last follows from (3) and Lemma 2.

Now, starting from Inequality (24) (which holds only for successful iterations), we can derive the following chain of inequalities

$$\begin{aligned}\psi_{p_i} - \psi_{p_i+1} &> v\,\mathrm{Ared}_k(x_{p_i+1}, \theta_{p_i+1}) - (1-v)(\delta_{p_i+1}^2 - \delta_{p_i}^2) \\ &\geq \frac{v\eta_1^2(1-r)\underline{h}}{g_{\max}\delta_{\max}}\delta_{p_i}\|g_{p_i}\| - (1-v)\frac{(\gamma^2-1)}{\eta_2}\delta_{p_i}\|g_{p_i}\| \\ &= \underbrace{\left(v\left(\frac{\eta_1^2(1-r)\underline{h}}{g_{\max}\delta_{\max}} + \frac{(\gamma^2-1)}{\eta_2}\right) - \frac{(\gamma^2-1)}{\eta_2}\right)}_{:=c}\delta_{p_i}\|g_{p_i}\|, \end{aligned} \tag{40}$$

where the second inequality follows from (39), (9), and (15). Now, recalling the definition of $v^\dagger$ given in (26), we choose v in (19) as

$$\max\left\{v^\dagger, \frac{(\gamma^2-1)g_{\max}\delta_{\max}}{\eta_1^2\eta_2(1-r)\underline{h} + (\gamma^2-1)g_{\max}\delta_{\max}}\right\} < v < 1, \tag{41}$$

and, consequently, c in (40) is positive while keeping Theorem 1 still applicable. Then, plugging $\|g_{p_i}\| \geq \frac{\epsilon}{2}$ into (40) yields

$$\psi_{p_i} - \psi_{p_i+1} > \frac{\epsilon c}{2}\delta_{p_i}.$$

Summing the previous inequality over $k \in \{p_i\}$, $k \geq \widetilde{k}$, and noting that $\psi_k - \psi_{k+1} > 0$ for any k (due to (27)), we obtain

$$\begin{aligned}\sum_{k\in\{p_i\},k\geq\widetilde{k}} \delta_k &< \frac{2}{\epsilon c}\sum_{k\in\{p_i\},k\geq\widetilde{k}}(\psi_k - \psi_{k+1}) \\ &\leq \lim_{K\to\infty}\frac{2}{\epsilon c}\sum_{k=\widetilde{k}}^{K}(\psi_k - \psi_{k+1}) \\ &= \lim_{k\to\infty}\frac{2}{\epsilon c}(\psi_{\widetilde{k}} - \psi_{K+1}) \\ &\leq \frac{2}{\epsilon c}\psi_{\widetilde{k}}, \end{aligned}$$

where the last inequality follows from (22). Then, we have shown that

$$\sum_{k\in\{p_i\}} \delta_k < \infty.$$

Furthermore, let us introduce the Bernoulli variable $B_k = 2 \cdot \mathbb{1}_{\mathcal{G}_{k,1}} \cdot \mathbb{1}_{\mathcal{G}_{k,2}} - 1$, which takes a value of 1 when the iteration k is true and a value of -1 otherwise. Note that due to Assumption 4,

$$\mathbb{P}(B_k = 1 | \mathcal{F}_{k-1}) > \frac{1}{2}.$$

Moreover, the sequence $\{\Delta_k\}$ is a sequence of non-negative uniformly bounded random variables. Then, we can proceed as in the proof in ([6], Lemma 4.2), and using ([6], Lemma 4.1) we obtain

$$\mathbb{P}\left(\left\{\sum_{k\in\{P_i\}}\Delta_k < \infty\right\} \cap \left\{\sum_{k\in\{N_i\}}\Delta_k = \infty\right\}\right) = 0.$$

This implies that almost surely

$$\sum_{k \in \{N_i\}} \Delta_k < \infty,$$

hence, the thesis follows. □

As a byproduct of the previous lemma, we obtain the expected convergence result in probability in the exact same way as in [6].

Theorem 3. *Suppose that Assumptions 1–4 hold. Let $\{X_k\}$ be the sequence of random iterates generated by Algorithm 1. Then, there holds*

$$\mathbb{P}\left(\lim_{k \to \infty} \|\nabla f(X_k)\| = 0\right) = 1.$$

Proof. The proof follows exactly as in ([6], Theorem 4.3). □

4. Numerical Illustration

In this section, we evaluate the numerical performance of Algorithm 1 equipped with the probabilistic accuracy requirements imposed in Assumption 4. Algorithm 1 was implemented using MATLAB R2019a, and the numerical experiments were performed on an 8 GB RAM laptop with an Intel Core i7-4510U CPU 2.00-2.60 GHz processor. The related software can be downloaded from sites.google.com/view/optml-italy-serbia/home/software (accessed on 1 September 2022).

We perform our numerical experiments on a binary classification problem. Denoting with $\{(a_i, b_i)\}_{i=1}^N$ a training set, where $a_i \in \mathbb{R}^n$ is the i-th feature vector, and $b_i \in \{0, 1\}$ is the associated label, we address the following nonconvex optimization problem:

$$\min_{x \in \mathbb{R}^n} f_N(x) = \frac{1}{N} \sum_{i=1}^{N} \left(b_i - \frac{1}{1 + e^{-a_i^T x}}\right)^2. \tag{42}$$

Note that (42) can be framed in Problem (1) by setting $\phi_i(x) = (b_i - 1/(1 + e^{-a_i^T x}))^2$, $i = 1, \ldots, N$, namely the composition of the least-square loss with the sigmoid function. Furthermore, it is easy to see that the objective function f_N satisfies Assumption 2 since each ϕ_i is continuously differentiable and f_N is bounded from below and above.

In Table 1, we report the four data sets used for our experiments. For each data set, we specify the number of feature vectors N, the number of components n of each feature vector, and the size N_T of the testing set I_{N_T}.

Table 1. Data sets used.

	Training Set		Testing Set
Data Set	N	n	N_T
A8A [26]	15,887	123	6809
A9A [26]	22,793	123	9768
MNIST [27]	60,000	784	10,000
PHISHING [28]	7739	68	3316

We implement two different versions of Algorithm 1, which differ from one another in the way the two sample sizes N_{k+1}^t and $N_{k+1,g}$ for the estimators in (4) and (5) are selected.

- **SIRTR$_{\mathbf{nop}}$**: this is Algorithm 1 implemented as in [11]. In particular, the infeasibility measure h and the initial penalty parameter θ_0 are chosen as follows:

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

In Step 1, the reference sample size $\widetilde{N}_{k+1}$ is computed as follows:

$$\widetilde{N}_{k+1} = \min\{N, \lceil \widetilde{c} N_k \rceil\}, \tag{43}$$

where $\widetilde{c} = 1.05$. It is easy to see that Rule (43) complies with Condition (6) by setting $r = (N - (\widetilde{c} - 1))/N$. In Step 2, the trial sample size N_{k+1}^t is chosen in compliance with Condition (7) as

$$N_{k+1}^t = \begin{cases} \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil \in [N_0, 0.95N] \\ \widetilde{N}_{k+1}, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil < N_0 \\ N, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil > 0.95N. \end{cases} \tag{44}$$

In Step 3, the sample size $N_{k+1,g}$ is fixed as

$$N_{k+1,g} = \lceil c N_{k+1}^t \rceil, \tag{45}$$

where $c = 0.1$. Furthermore, the set $I_{N_{k+1}^t}$ for computing $f_{N_{k+1}^t}(x_k)$ and $f_{N_{k+1}^t}(x_k + p_k)$ is sampled uniformly at random using the MATLAB command `randsample`, whereas $g_k \in \mathbb{R}^n$ is a sample average approximation as in (5) using $I_{N_{k+1,g}} \subseteq I_{N_{k+1}^t}$. The other parameters are set as $x_0 = (0, 0, \ldots, 0)^T$, $\delta_0 = 1$, $\delta_{\max} = 1$, $\gamma = 2$, $\eta = 10^{-1}$, $\eta_2 = 10^{-6}$, $\mu = 100/N$. Note that Choices (44)–(45) for the sample sizes $N_{k+1}^t, N_{k+1,g}$ are not sufficient to guarantee that Assumption 4 holds so that Theorems 2–3 do not apply to this version of Algorithm 1.

- **SIRTR$_\mathbf{p}$**: this implementation of Algorithm 1 differs from the previous one only in the choice of the sample sizes $N_{k+1}^t, N_{k+1,g}$. In this case, we force these two parameters to comply with Assumption 4. According to ([29], Theorem 7.2, Table 7.1), a subsampled estimator $\nabla f_S(x_k) = \frac{1}{S} \sum_{i \in I_S} \nabla \phi_i(x_k)$ with sample size $|I_S| = S$ satisfies the probabilistic requirement

$$\mathbb{P}(\|\nabla f_S(X_k) - \nabla f_N(X_k)\| \leq \nu \delta_k | \mathcal{F}_{k-1}) \geq \pi, \tag{46}$$

where $\pi \in (0, 1)$ if the sample size S complies with the following lower bound

$$S \geq N_{k+1}^{\chi,\nu,\pi} = \min\left\{N, \left\lceil \frac{4\chi}{\nu\delta_k}\left(\frac{2\chi}{\nu\delta_k} + \frac{1}{3}\right)\log\left(\frac{n+1}{1-\pi}\right)\right\rceil\right\}, \tag{47}$$

where $\chi = \frac{1}{5}\max_{i=1,\ldots,N}\|a_i\|$. Based on the previous remark, we choose the samples sizes of SIRTR$_\mathrm{p}$ as follows

$$N_{k+1}^t = \begin{cases} \max\{N_{k+1}^{\chi,\nu,\pi}, \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil\}, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil \in [N_0, 0.95N] \\ \max\{N_{k+1}^{\chi,\nu,\pi}, \widetilde{N}_{k+1}\}, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil < N_0 \\ N, & \text{if } \lceil \widetilde{N}_{k+1} - \mu N \delta_k^2 \rceil > 0.95N. \end{cases} \tag{48}$$

$$N_{k+1,g} = \max\left\{N_{k+1}^{\chi,\nu,\pi}, \lceil c N_{k+1}^t \rceil\right\}. \tag{49}$$

Setting $\pi > 1/\sqrt{2}$, choosing $N_{k+1}^t, N_{k+1,g}$ as in (48) and (49), and sampling $I_{N_{k+1}^t}$ and $I_{N_{k+1,g}}$ uniformly and independently at random in $\{1, \ldots, N\}$, we guarantee that Assumption 4 holds with $\pi_1 = \pi_2 = \pi$, thus ensuring the convergence in probability of SIRTR$_\mathrm{p}$ according to Theorems 2–3. For our tests, we set $\pi = 3/4$ and $\nu = 10\|\nabla f_N(x_0) - \nabla f_{N_0}(x_0)\|$.

For each data set, we perform 10 runs of both SIRTR$_\mathrm{nop}$ and SIRTR$_\mathrm{p}$ and assess their performances by measuring the following metrics:

- **training loss**, given as $f_N(x_k)$ with f_N defined in (42);

- **testing loss**, defined as

$$f_{N_T}(x_k) = \frac{1}{N_T} \sum_{i \in I_{N_T}} \phi_i(x_k),$$

where I_{N_T} denotes the testing set and N_T its dimension;
- **classification error**, defined as

$$e_k = \frac{1}{N_T} \sum_{i \in I_{N_T}} |b_i - b_i^{pred}(x_k)|, \tag{50}$$

where b_i denotes the true label of the $i-$th feature vector of the testing set and $b_i^{pred}(x_k) = \max\{\text{sign}(a_i^T x_k), 0\}$ is the corresponding predicted label at iteration k.

We note that (42) can be seen as the optimization problem arising from training a neural network with no hidden layers and the sigmoid function as the activation function for the output layer. Then, as in [30,31], we measure the computational cost of evaluating the objective function and its gradient in terms of forward and backward propagations. Namely, we count the number of full function and gradient evaluations, by considering the computation of a single function ϕ_i equivalent to $\frac{1}{N}$ forward propagations, and the evaluation of a single gradient $\nabla\phi_i$ equivalent to $\frac{2}{N}$ propagations. Regarding SIRTR$_{\text{nop}}$, we note that the computational cost per iteration is determined by $\frac{N_{k+1}^t + N_{k+1,g}}{N}$ propagations since $I_{N_{k+1,g}} \subseteq I_{N_{k+1}^t}$. On the contrary, the computational cost of SIRTR$_{\text{p}}$ is determined by $\frac{N_{k+1}^t}{N} + \frac{2N_{k+1,g}}{N}$ propagations, as $I_{N_{k+1}^t}$ and $I_{N_{k+1,g}}$ are sampled independently from one another. For both algorithms, the computational cost per iteration increases as the iterations proceed; indeed, since δ_k tends to zero as k tends to infinity (Theorem 1), Rules (44)–(48) will eventually select the trial sample size N_{k+1}^t equal to the reference sample size $\widetilde{N}_{k+1}$, which is increasing geometrically. We expect that the computational cost increases faster in SIRTR$_{\text{p}}$, as this algorithm also requires the gradient sample size $N_{k+1,g}$ to increase due to Conditions (47)–(49). Finally, we note that the computational cost per iteration of both algorithms is higher than that of the standard stochastic gradient algorithm, which is usually $\frac{2N_g}{N}$, with N_g being a prefixed gradient sample size. However, the increasing sample sizes result in more accurate function and gradient approximations so the higher computational cost likely implies a larger reduction in the training loss f_N per iteration, as seen in previous comparisons of SIRTR with a non-adaptive stochastic approach in [11].

In Figure 1, we show the decrease in the training loss, testing loss, and classification error (all averaged over the 10 runs) versus the average computational cost for the first 20 propagations. For most data sets, we observe that SIRTR$_{\text{nop}}$ performs comparably or even better than SIRTR$_{\text{p}}$. However, for one of the four data sets (`mnist`) in SIRTR$_{\text{nop}}$, the accuracy deteriorates after the first propagations, whereas SIRTR$_{\text{p}}$ provides a more accurate classification and a quite steady decrease in the average training loss, testing loss, and classification error. This different performance of the two algorithms can be explained by looking at Figure 2, which shows the increase in the percentage ratio $\frac{100N_{k+1}}{N}$ of the sample size N_{k+1} over the data set size N (averaged over the 10 runs) for both algorithms. As we can see, the sample size in SIRTR$_{\text{p}}$ rapidly increases to 60% of the data set size in the first 50 iterations, whereas the same percentage is achieved by SIRTR$_{\text{nop}}$ after 150–200 iterations. Overall, we can conclude that the probabilistic requirements of Assumption 4 ensure the theoretical support for convergence in probability but might be excessively demanding. In fact, the numerical examples show that a slower increase in the sample size than that imposed by the probabilistic requirements of Assumption 4 provides a good trade-off between the computational cost and the accuracy in the classification.

In Figure 3, we test the sensitivity of SIRTR$_{\text{p}}$ with respect to the initial penalty parameter θ_0 by reporting the average training loss versus the average computational cost obtained with three different values of θ_0. We observe that the performance of the algorithm

is not considerably affected by the choice of this parameter, although large oscillations in the average training loss occur for smaller values of θ_0 in `mnist` when $\theta_0 = 0.1$. As a general comment, small initial values of θ_0 may not be convenient, as the sequence $\{\theta_k\}$ is non-increasing and small values of θ_k promote a decrease in the infeasibility measure h rather than a decrease in the training loss (see the definition of the actual reduction in (13)). Similar considerations can be made for SIRTR$_{\text{nop}}$ in Figure 4.

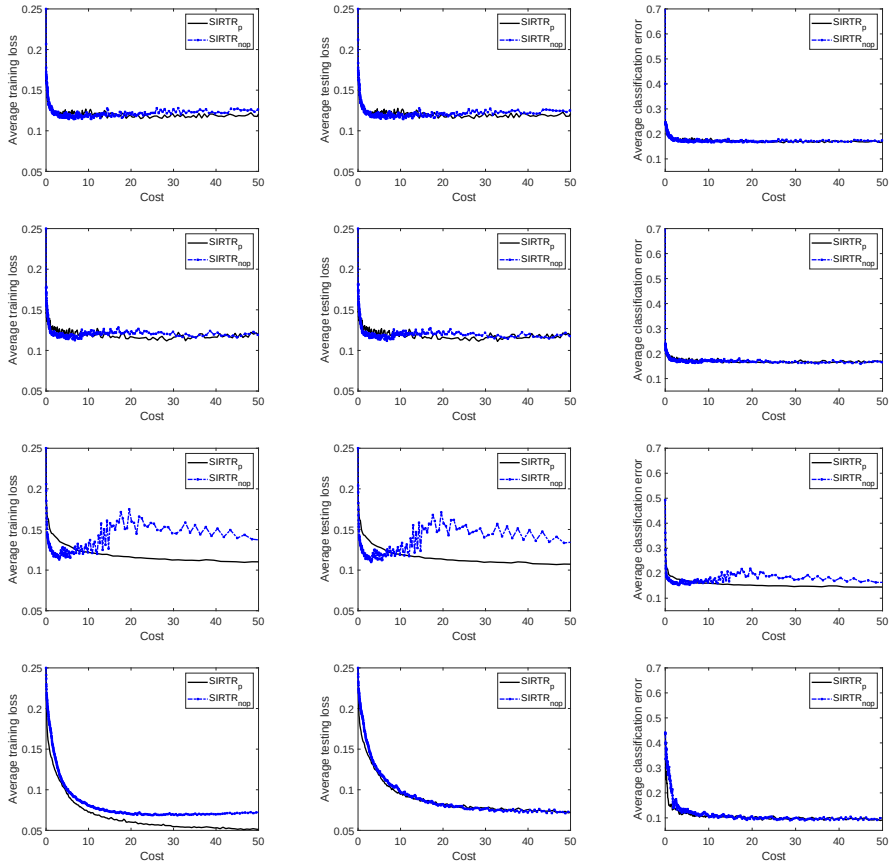

Figure 1. From top to bottom row: data sets `a8a`, `a9a`, `mnist`, `phishing`. From left to right: average training loss, testing loss, and classification error versus average computational cost.

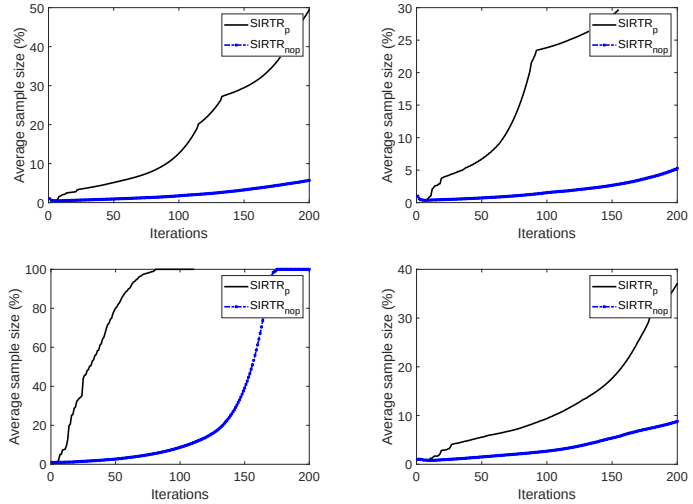

Figure 2. Average percentage ratio of the sample size N_{k+1} over the data set size N versus iterations. **Top row**: `a8a` (**left**) and `a9a` (**right**). **Bottom row**: `mnist` (**left**) and `phishing` (**right**).

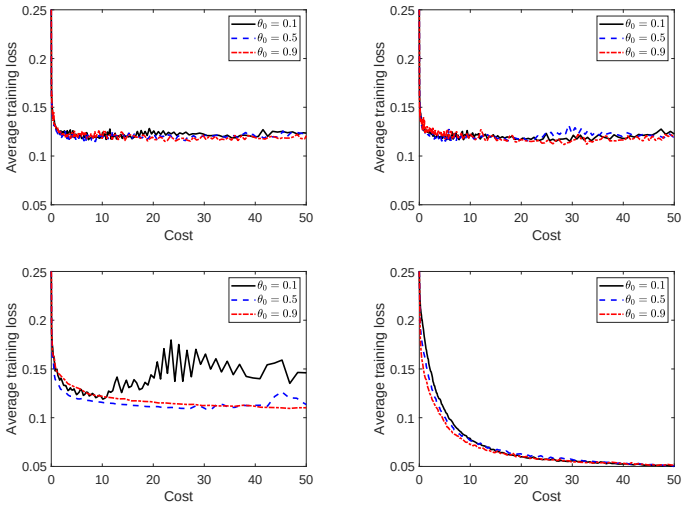

Figure 3. Average training loss versus average computational cost of SIRTR$_{\mathrm{p}}$ equipped with different values for the initial penalty parameter θ_0. **Top row**: `a8a` (**left**) and `a9a` (**right**). **Bottom row**: `mnist` (**left**) and `phishing` (**right**).

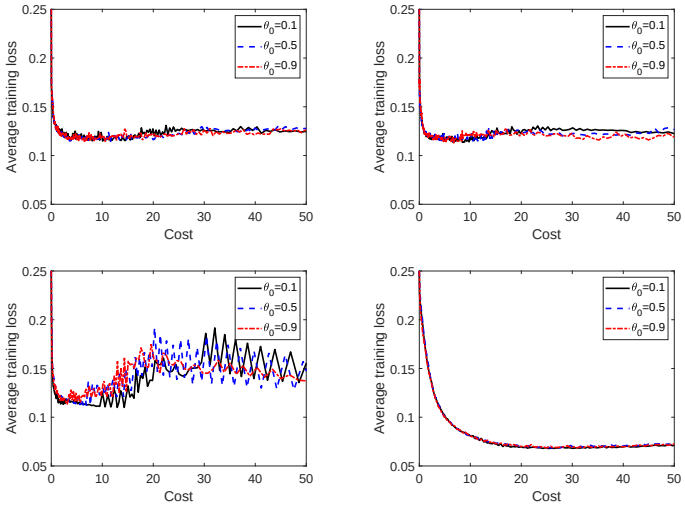

Figure 4. Average training loss versus average computational cost of SIRTR$_{\mathrm{nop}}$ equipped with different values for the initial penalty parameter θ_0. **Top row**: `a8a` (**left**) and `a9a` (**right**). **Bottom row**: `mnist` (**left**) and `phishing` (**right**).

5. Conclusions

In this paper, we have proved the convergence in probability of a stochastic trust-region method based on the inexact restoration approach (SIRTR) under the assumption that the function and gradient estimates are sufficiently accurate with sufficiently high probability. This result is novel for SIRTR and agrees with other results obtained in the existing literature [16,18,19]. The numerical experiments on binary classification show that the probabilistic requirements improve the numerical stability of the algorithm, ensuring satisfactory accuracy for all data sets. Future work could address the replacement of the probabilistic requirements considered here with alternative strategies for ensuring convergence in probability, such as variance reduction techniques, or the development of a second-order version of SIRTR for improving accuracy based on approximations of the Hessian obtained through subsampling.

Author Contributions: Conceptualization, S.B. and B.M. and S.R.; methodology, S.B. and B.M. and S.R.; software, S.B. and B.M. and S.R.; validation, S.B. and B.M. and S.R.; writing—original draft preparation, S.B. and B.M. and S.R.; writing—review and editing, S.B. and B.M. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the INdAM GNCS project "Ottimizzazione adattiva per il machine learning" (CUP_E55F22000270001) and the Mobility Project "Second order methods for optimization problems in Machine Learning" (ID: RS19MO05) within the frame of the executive Program of Cooperation in the Field of Science and Technology between the Italian Republic and the Republic of Serbia 2019–2022. The third author also acknowledges the financial support received from the IEA CNRS project entitled "VaMOS".

Data Availability Statement: The data sets employed in this paper have been accessed on 1 September 2022 at the following repositories: http://www.csie.ntu.edu.tw/~cjlin/libsvm, http://yann.lecun.com/exdb/mnist, https://archive.ics.uci.edu/ml/index.php (accessed on 1 September 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
2. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
3. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* **2018**, *60*, 223–311. [CrossRef]
4. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
5. Schmidt, M.; Le Roux, N.; Bach, F. Minimizing Finite Sums with the Stochastic Average Gradient. *Math. Program.* **2017**, *162*, 83–112. [CrossRef]
6. Bandeira, A.S.; Scheinberg, K.; Vicente, L.N. Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.* **2014**, *24*, 1238–1264. [CrossRef]
7. Bellavia, S.; Gurioli, G.; Morini, B.; Toint, P.L. Adaptive regularization algorithm for nonconvex optimization using inexact function evaluations and randomly perturbed derivatives. *J. Complex.* **2022**, *68*, 101591. [CrossRef]
8. 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.* **2022**, *10*, 100043. [CrossRef]
9. Bellavia, S.; Krejić, N.; Krklec Jerinkic, N. Subsampled Inexact Newton methods for minimizing large sums of convex functions. *IMA J. Numer. Anal.* **2020**, *40*, 2309–2341. [CrossRef]
10. Bellavia, S.; Krejić, N.; Morini, B. Inexact restoration with subsampled trust-region methods for finite-sum minimization. *Comp. Opt. Appl.* **2020**, *73*, 701–736. [CrossRef]
11. Bellavia, S.; Krejić, N.; Morini, B.; Rebegoldi, S. A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. *Comput. Optim. Appl.* **2022**. [CrossRef]
12. Bergou, E.; Gratton, S.; Vicente, L.N. Levenberg–Marquardt Methods Based on Probabilistic Gradient Models and Inexact Subproblem Solution, with Application to Data Assimilation. *SIAM-ASA J. Uncertain.* **2016**, *4*, 924–951. [CrossRef]
13. Bergou, E.H.; Diouane, Y.; Kunc, V.; Kungurtsev, V.; Royer, C.W. A subsampling line-search method with second-order results. *INFORMS J. Optim.* **2022**, *4*, 403–425. [CrossRef]
14. Bollapragada, R.; Byrd, R.; Nocedal, J. Adaptive sampling strategies for stochastic optimization. *SIAM J. Optim.* **2018**, *28*, 3312–3343. [CrossRef]
15. Bollapragada, R.; Byrd, R. Nocedal, Exact and Inexact Subsampled Newton Methods for Optimization. *IMA J. Numer. Anal.* **2019**, *9*, 545–578. [CrossRef]
16. Chen, R.; Menickelly, M.; Scheinberg, K. Stochastic optimization using a trust-region method and random models. *Math. Program.* **2018**, *169*, 447–487. [CrossRef]
17. di Serafino, D.; Krejic, N.; Krklec Jerinkic, N.; Viola, M. LSOS: Line-search Second-Order Stochastic optimization methods for nonconvex finite sums. *arXiv* **2021**, arXiv:2007.15966v2.
18. Larson, J.; Billups, S.C. Stochastic derivative-free optimization using a trust region framework. *Comput. Optim. Appl.* **2016**, *64*, 619–645. [CrossRef]
19. Paquette, C.; Scheinberg, K. A Stochastic Line Search Method with Expected Complexity Analysis. *SIAM J. Optim.* **2020**, *30*, 349–376. [CrossRef]
20. Xu, P.; Roosta-Khorasani, F.; Mahoney, M.W. Newton-Type Methods for Non-Convex Optimization Under Inexact Hessian Information. *Math. Program.* **2020**, *184*, 35–70. [CrossRef]
21. Blanchet, J.; Cartis, C.; Menickelly, M.; Scheinberg, K. Convergence Rate Analysis of a Stochastic Trust Region Method via Submartingales. *INFORMS J. Optim.* **2019**, *1*, 92–119. [CrossRef]

22. Cartis, C.; Scheinberg, K. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Program.* **2018**, *169*, 337–375. [CrossRef]
23. Birgin, E.G.; Krejić, N.; Martínez, J.M. Inexact restoration for derivative-free expensive function minimization and applications. *J. Comput. Appl. Math.* **2022**, *410*, 114193. [CrossRef]
24. Bueno, L.F.; Friedlander, A.; Martínez, J.M.; Sobral, F.N.C. Inexact Restoration Method for Derivative-Free Optimization with Smooth Constraints. *SIAM J. Optim.* **2013**, *23*, 1189–1213. [CrossRef]
25. Martínez, J.M. Pilotta, Inexact restoration algorithms for constrained optimization. *J. Optim. Theory Appl.* **2000**, *104*, 135–163. [CrossRef]
26. Lichman, M. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 1 September 2022).
27. LeCun, Y.; Bottou, L.; Bengio, Y. Haffner, The MNIST Database. Available online: http://yann.lecun.com/exdb/mnist (accessed on 1 September 2022).
28. Chang, C.C.; Lin, C.J. LIBSVM: A Library for Support Vector Machines. Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm (accessed on 1 September 2022).
29. Bellavia, S.; Gurioli, G.; Morini, B.; Toint, P.L. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM J. Optim.* **2019**, *29*, 2881–2915. [CrossRef]
30. Bellavia, S.; Gurioli, G. Stochastic analysis of an adaptive cubic regularization method under inexact gradient evaluations and dynamic Hessian accuracy. *Optimization* **2022**, *71*, 227–261. [CrossRef]
31. 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, Cincinnati, OH, USA, 7–9 May 2020.