This is the peer reviewd version of the followng article:

An investigation of stochastic trust-region based algorithms for finite-sum minimization / Bellavia, S., Morini, B., Rebegoldi, S.. - In: OPTIMIZATION METHODS & SOFTWARE. - ISSN 1055-6788. - (2024), pp. 1-30. [10.1080/10556788.2024.2346834]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

11/10/2026 16:41

(Article begins on next page)

arXiv:2404.13382v1 [math.OC] 20 Apr 2024

AN INVESTIGATION OF STOCHASTIC TRUST-REGION BASED ALGORITHMS FOR FINITE-SUM MINIMIZATION

STEFANIA BELLAVIA*, BENEDETTA MORINI*, SIMONE REBEGOLDI†

Abstract. This work elaborates on the TRust-region-ish (TRish) algorithm, a stochastic optimization method for finite-sum minimization problems proposed by Curtis et al. in [15,16]. A theoretical analysis that complements the results in the literature is presented, and the issue of tuning the involved hyper-parameters is investigated. Our study also focuses on a practical version of the method, which computes the stochastic gradient by means of the inner product test and the orthogonality test proposed by Bollapragada et al. in [7]. It is shown experimentally that this implementation improves the performance of TRish and reduces its sensitivity to the choice of the hyper-parameters.

1. Introduction.

We consider the finite-sum minimization problem

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

where N is a large integer and each function F_i, $1 \le i \le N$, is differentiable, and we address its solution by the stochastic first-order method named TRust-regionish-algorithm (TRish) and presented in [15].

TRish combines ideas from the Stochastic Gradient (SG) method [21] and the trust-region methodology [13]. Depending on the value of the norm of the stochastic gradient, the step takes either the form of the SG step or the form of a normalized step inspired by a first-order trust-region scheme. This strategy aims at reducing the dependence of the steplength from the specific function F, and as a result, TRish can outperform a traditional SG algorithm from the numerical viewpoint, see the experiments in [15]. An extension of TRish to second-order models was recently proposed in [16].

TRish differs from the stochastic trust-region methods in [1–4,6,10,12,14] in two main respects. First, the computed step is accepted at each iteration, namely it is not ruled by the actual-to-predicted reduction ratio. Second, the theoretical properties of TRish are obtained by assuming an upper bound on the variance of the stochastic gradient estimates, which is common in the analysis of the SG method, whereas specific accuracy requirements on the random model at each iteration are not required.

Our interest in TRish derives from the experimental comparison we carried out among various stochastic trust-region methods. In our experience, after tuning the parameters, TRish compares well with adaptive stochastic trust-region methods, see [4]. Limiting our attention to the first-order version of TRish, we observe that the computation of the step at each iteration requires three hyper-parameters, as well as a rule for computing the stochastic gradient. These hyper-parameters are the steplength and two positive scalars employed in the choice between a SG-type step and a normalized step; these parameters can vary along the iterations. Due to the form of

*Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale G.B. Morgagni 40, 50134 Firenze, Italia. Members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INDAM). Emails: stefania.bellavia@unifi.it, benedetta.morini@unifi.it

†Dipartimento di Scienze Fisiche, Informatiche e Matematiche, Università degli Studi di Modena e Reggio Emilia, Via Campi 213/B, 41125 Modena, Italia. Member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INDAM). Email: simone.rebegoldi@unimore.it

‡The research that led to the present paper was partially supported by INDAM-GNCS through Progetti di Ricerca 2023 and by PNRR - Missione 4 Istruzione e Ricerca - Componente C2 Investimento 1.1, Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN) funded by the European Commission under the NextGeneration EU programme, project "Advanced optimization METhods for automated central veIn Sign detection in multiple sclerosis from magneTic resonAnce imaging (AMETISTA)", code: P2022J9SNP, MUR D.D. financing decree n. 1379 of 1st September 2023 (CUP E53D23017980001), project "Numerical Optimization with Adaptive Accuracy and Applications to Machine Learning", code: 2022N3ZNAX MUR D.D. financing decree n. 973 of 30th June 2023 (CUP B53D23012670006), and project "Inverse problems in PDE: theoretical and numerical analysis", code: 2022B32J5C MUR D.D. financing decree n. 973 of 30th June 2023 (CUP B53D23009200006).

problem (1.1), the stochastic gradient can be naturally built using a mini-batch approach, which requires the selection of a sample size at each iteration. Note that the sample size is implicitly a further hyper-parameter of the algorithm. Clearly, the performance of TRish depends on the hyper-parameters choice, and its application requires tuning them for each problem, analogously to SG methods. This tuning process can be computationally onerous, and represents the main limitation of the algorithm.

In this paper, we pursue the analysis of TRish, by focusing on its theoretical properties and the practical choice of both the hyper-parameters and the batch size used for computing the gradient estimator. We study the theoretical properties of the first-order version of the algorithm, under the assumption that the three hyper-parameters above-mentioned are constant, and provide results that complement the convergence analysis in [15, 16], in terms of both conditions on such parameters and expected decrease of the optimality measure. Furthermore, inspired by [7], we investigate the use of adaptive sampling techniques that vary dynamically the sample size for computing the stochastic gradient, and provide a practical implementation of TRish that complies with the theoretical assumptions of our analysis. We show experimentally that an adaptive strategy for building the stochastic gradient reduces the dependence of the performance of TRish from its parameters, although it does not completely overcome the need to tune them.

The paper is structured as follows. In section 2, we briefly recall TRish and explore its convergence properties under different settings. In section 3, we present an implementation of the algorithm, named TRish_AS, which is based on the adaptive sampling techniques proposed in [7]. In section 4, we report the numerical performance of TRish_AS on a series of classification and regression problems. Some conclusions are offered in section 5.

Notation. The symbol $\mathbb{E}_k[\cdot]$ denotes the conditional expectation of a random variable conditioned on the event that the $k-$th iterate of the algorithm is x_k. The symbol $\|\cdot\|$ denotes the Euclidean norm.

2. The algorithm and convergence results. TRish is a simple yet effective algorithm for stochastic and finite-sum minimization problems [15]. At each iteration k, a stochastic estimate g_k of the gradient is formed and used for building the step. A normalized step along g_k is built whenever the norm of g_k lies within a specified interval $\left[\frac{1}{\gamma_{1,k}}, \frac{1}{\gamma_{2,k}}\right]$, otherwise the step is of SG-type. The stepsize depends on a given positive scalar α_k. In [15, 16], convergence results have been given for constant sequences $\{\alpha_k\}$, $\{\gamma_{1,k}\}$, $\{\gamma_{2,k}\}$, as well as for diminishing stepsizes. Here, we focus on the case where the parameters sequences are constant. The TRish algorithm is given below.

Algorithm 1 TRish

Choose an initial iterate $x_0 \in \mathbb{R}^n$.
Choose a steplength $\alpha > 0$ and parameters $0 < \gamma_2 < \gamma_1$.
Set $k = 0$.
Repeat until a convergence test is satisfied

1. Generate a stochastic gradient $g_k \approx \nabla F(x_k)$.
2. Compute

$$p_k = -\begin{cases} \gamma_1 \alpha g_k, & \text{if } \|g_k\| \in [0, \frac{1}{\gamma_1}) \\ \alpha \dfrac{g_k}{\|g_k\|}, & \text{if } \|g_k\| \in [\frac{1}{\gamma_1}, \frac{1}{\gamma_2}] \\ \gamma_2 \alpha g_k, & \text{if } \|g_k\| \in (\frac{1}{\gamma_2}, +\infty[\end{cases}, \qquad x_{k+1} = x_k + p_k. \tag{2.1}$$

3. Set $k = k + 1$.

Algorithm 1 is based on a *trust-regionish* procedure. The step p_k takes different forms depending on the three cases in (2.1). In the second case, $\|g_k\| \in [\frac{1}{\gamma_1}, \frac{1}{\gamma_2}]$, the step taken solves the

trust-region problem

$$\begin{aligned} \min_{p \in \mathbb{R}^n} \quad & F(x_k) + g_k^T p. \\ \|p\| \leq \alpha \end{aligned}$$

In contrast to deterministic and stochastic trust-region methodologies [1–4, 6, 10, 12–14], the adopted step may differ from the trust-region step and it is accepted at each iteration, namely no acceptance rules are tested. The reason for choosing between a trust-region step and a SG-type step derives from the observation that using the trust-region step $p_k = -\alpha g_k/\|g_k\|$ at each iteration can prevent the algorithm from making progress in expectation, see [15, p. 202].

In the following, we provide convergence results for TRish that complement those given in [15,16]. We make the following assumptions on F.

ASSUMPTION 2.1. *F is continuously differentiable and bounded from below by $F_* = \inf_{x\in\mathbb{R}^n} F(x)$. The gradient $\nabla F : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with constant $L > 0$, i.e.,*

$$\|\nabla F(x) - \nabla F(y)\| \leq L\|y - x\|, \quad \forall\ x, y \in \mathbb{R}^n. \tag{2.2}$$

Results are presented for both general nonconvex objective functions F and for functions satisfying the Polyak-Lojasiewicz (PL) condition stated below.

ASSUMPTION 2.2. *There exists a constant $\mu > 0$ such that*

$$\|\nabla F(x)\|^2 \geq 2\mu(F(x) - F_*), \quad \forall\ x \in \mathbb{R}^n. \tag{2.3}$$

Strongly convex functions and strongly convex functions composed with linear maps satisfy the PL condition [18]. We note that $L \geq \mu$, see e.g. [9, pag. 246].

As one would expect, the analysis of TRish heavily relies on the assumptions made on the stochastic gradient g_k. In the remainder of this section, we make the following assumption.

ASSUMPTION 2.3. *The vector g_k is an unbiased estimator of $\nabla F(x_k)$, i.e.,*

$$\mathbb{E}_k[g_k] = \nabla F(x_k). \tag{2.4}$$

Under this assumption, $(-g_k)$ is a descent direction in expectation since

$$\mathbb{E}_k[g_k^T \nabla F(x_k)] = \|\nabla F(x_k)\|^2 > 0, \tag{2.5}$$

and consequently

$$\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2] = -\|\nabla F(x_k)\|^2 + \mathbb{E}_k[\|g_k\|^2]. \tag{2.6}$$

Before presenting our analysis, we give two simple inequalities that will be repeatedly used

$$\begin{aligned} 0 \leq \frac{1}{2}\left(\|g_k\| - \|\nabla F(x_k) - g_k\|\right)^2 &= \frac{1}{2}\|g_k\|^2 - \|\nabla F(x_k) - g_k\|\|g_k\| + \\ &\frac{1}{2}\|\nabla F(x_k) - g_k\|^2, \end{aligned} \tag{2.7}$$

$$\begin{aligned} 0 \leq \frac{\gamma_2}{2\gamma_1^2}\left(1 - \frac{\gamma_1^2}{\gamma_2}\|\nabla F(x_k) - g_k\|\right)^2 &= \frac{\gamma_2}{2\gamma_1^2} - \|\nabla F(x_k) - g_k\| + \\ &\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2, \end{aligned} \tag{2.8}$$

and we introduce a positive scalar that will appear in the results

$$\beta = \frac{\gamma_1^2 - \gamma_2^2}{2\gamma_2} + \frac{1}{2}\alpha\gamma_1^2 L. \tag{2.9}$$

Note that β is positive as $\gamma_2 < \gamma_1$.

The first result provides bounds on the expected decrease of the objective function yielded by TRish from one iteration to the other.

LEMMA 2.1. *Let Assumptions 2.1 and 2.3 hold. For all k, it holds*

$$\begin{aligned}
\mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k)\|^2 + \alpha\beta\mathbb{E}_k[\|g_k\|^2], &(2.10)\\
\mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \frac{1}{2}\alpha\left(\gamma_2 - \alpha\gamma_1^2 L\right)\|\nabla F(x_k)\|^2 + \\
&\quad \alpha\beta\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2], &(2.11)
\end{aligned}$$

with β given in (2.9).

Proof. By applying the Descent Lemma for functions with a Lipschitz continuous gradient [5, Proposition A.24], we obtain

$$F(x_{k+1}) \leq F(x_k) + \nabla F(x_k)^T p_k + \frac{L}{2}\|p_k\|^2, \tag{2.12}$$

and it trivially follows

$$F(x_{k+1}) \leq F(x_k) + g_k^T p_k + (\nabla F(x_k) - g_k)^T p_k + \frac{1}{2}L\|p_k\|^2. \tag{2.13}$$

Then, we show that, for all k, it holds

$$F(x_{k+1}) \leq F(x_k) - \frac{1}{2}\alpha(\gamma_2 - \alpha\gamma_1^2 L)\|g_k\|^2 + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2. \tag{2.14}$$

To this aim, we prove that inequality (2.14) holds irrespectively of the form of p_k, namely it holds for Case 1 ($p_k = -\gamma_1\alpha g_k$), Case 2 ($p_k = -\alpha g_k/\|g_k\|$), and Case 3 ($p_k = -\gamma_2\alpha g_k$).

In Case 1, we have $p_k = -\gamma_1\alpha g_k$ and equations (2.13), (2.7) give

$$\begin{aligned}
F(x_{k+1}) &\leq F(x_k) - \alpha\gamma_1\|g_k\|^2 - \alpha\gamma_1(\nabla F(x_k) - g_k)^T g_k + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2\\
&\leq F(x_k) - \alpha\gamma_1\|g_k\|^2 + \alpha\gamma_1\|\nabla F(x_k) - g_k\|\|g_k\| + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2\\
&\leq F(x_k) - \alpha\gamma_1\|g_k\|^2 + \frac{1}{2}\alpha\gamma_1\|g_k\|^2 + \frac{1}{2}\alpha\gamma_1\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2\\
&= F(x_k) - \frac{1}{2}\alpha\gamma_1\|g_k\|^2 + \frac{1}{2}\alpha\gamma_1\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2\\
&\leq F(x_k) - \frac{1}{2}\alpha\gamma_2\|g_k\|^2 + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2,
\end{aligned}$$

where the last inequality follows from $-\gamma_1 < -\gamma_2$ and $\gamma_1/\gamma_2 > 1$.

In Case 2, we have $p_k = -\alpha g_k/\|g_k\|$ and equations (2.13), (2.8) give

$$\begin{aligned}
F(x_{k+1}) &\leq F(x_k) - \alpha\|g_k\| - \alpha(\nabla F(x_k) - g_k)^T\frac{g_k}{\|g_k\|} + \frac{1}{2}\alpha^2 L\\
&\leq F(x_k) - \alpha\|g_k\| + \alpha\|\nabla F(x_k) - g_k\| + \frac{1}{2}\alpha^2 L\\
&\leq F(x_k) - \alpha\|g_k\| + \alpha\frac{\gamma_2}{2\gamma_1^2} + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2 L.
\end{aligned}$$

By noting that $\gamma_1\|g_k\| \geq 1$ and $\gamma_2\|g_k\| \leq 1$, it follows

$$\begin{aligned}
F(x_{k+1}) &\leq F(x_k) - \alpha\gamma_2\|g_k\|^2 + \alpha\frac{\gamma_2}{2\gamma_1^2}\gamma_1^2\|g_k\|^2 + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2\\
&\leq F(x_k) - \frac{1}{2}\alpha\gamma_2\|g_k\|^2 + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2.
\end{aligned}$$

In Case 3, we have $p_k = -\gamma_2 \alpha g_k$, and proceeding as in Case 1, equations (2.13), (2.7) give

$$\begin{aligned} F(x_{k+1}) &\leq F(x_k) - \frac{1}{2}\alpha\gamma_2\|g_k\|^2 + \frac{1}{2}\alpha\gamma_2\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_2^2 L\|g_k\|^2 \\ &\leq F(x_k) - \frac{1}{2}\alpha\gamma_2\|g_k\|^2 + \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k) - g_k\|^2 + \frac{1}{2}\alpha^2\gamma_1^2 L\|g_k\|^2, \end{aligned}$$

where the last inequality follows from $\gamma_2 < \gamma_1$ and $\gamma_1/\gamma_2 > 1$.

Finally, equations (2.14) and (2.6) yield

$$\begin{aligned} \mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \frac{1}{2}\alpha(\gamma_2 - \alpha\gamma_1^2 L)\mathbb{E}[|g_k\|^2] + \alpha\frac{\gamma_1^2}{2\gamma_2}\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2] \\ &= F(x_k) - \frac{1}{2}\alpha(\gamma_2 - \alpha\gamma_1^2 L)\mathbb{E}[|g_k\|^2] + \alpha\frac{\gamma_1^2}{2\gamma_2}(-\|\nabla F(x_k)\|^2 + \mathbb{E}_k[\|g_k\|^2]) \\ &= F(x_k) - \alpha\frac{\gamma_1^2}{2\gamma_2}\|\nabla F(x_k)\|^2 + \alpha\left(\frac{\gamma_1^2 - \gamma_2^2}{2\gamma_2} + \frac{1}{2}\alpha\gamma_1^2 L\right)\mathbb{E}_k[\|g_k\|^2], \end{aligned}$$

which gives (2.10) due to the definition (2.9) of β. Furthermore, equations (2.14) and (2.6) give

$$\begin{aligned} \mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \frac{1}{2}\alpha(\gamma_2 - \alpha\gamma_1^2 L)\mathbb{E}[\|g_k\|^2] + \alpha\frac{\gamma_1^2}{2\gamma_2}\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2] \\ &= F(x_k) - \frac{1}{2}\alpha(\gamma_2 - \alpha\gamma_1^2 L)(\|\nabla F(x_k)\|^2 + \mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2]) + \\ &\quad \alpha\frac{\gamma_1^2}{2\gamma_2}\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2], \end{aligned}$$

and (2.11) follows from the definition (2.9) of β. □

With respect to the existing literature, the arguments used above are simpler than those used in [15, Lemma 1] for deriving an analogous result. Furthermore, comparing (2.10) and (2.11) with the claim of [15, Lemma 1], we do not involve either the event $\nabla F(x_k)^T g_k \geq 0$, nor its probability.

The second result shows the behaviour of the expected decrease of the optimality gap $F(x_{k+1}) - F_*$ under the assumption that F satisfies the PL condition, as well as the behaviour of the expected decrease of the average square norm of the gradient $\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(x_k)\|^2\right]$ in case F is a general nonconvex function. Our analysis is carried out under the following assumption on the stochastic gradient.

ASSUMPTION 2.4. *There exists a positive constant M_g such that, for all k,*

$$\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2] < M_g.$$

This assumption was supposed to hold in the analysis of the second-order version of TRish carried out in [16]. It states that the variance of g_k is uniformly bounded and it can be nonzero at any stationary point of F. Such assumption includes the case where g_k is a stochastic approximation of $\nabla F(x_k)$ with additive noise, i.e., $g_k = \frac{1}{N}\sum_{i=1}^{N}(\nabla F_i(x_k) + \omega_i)$, each $\omega_i \in \mathbb{R}^n$ has mean zero and each component $(\omega_i)_j$ of ω_i has limited variance*, i.e., $\mathbb{E}_k[\omega_i] = 0$, $\mathbb{E}_k[(\omega_i)_j^2] \leq \sigma$, for some $\sigma > 0$, $1 \leq i \leq N$, $1 \leq j \leq n$. In addition, Assumption 2.4 holds when g_k is un unbiased gradient estimate built by sampling, i.e., $g_k = \frac{1}{|S_k|}\sum_{i\in S_k}\nabla F_i(x_k)$ with $S_k \subseteq \{1, \ldots, N\}$ and $|S_k|$ denoting the cardinality of the set S_k. If the variance of each stochastic gradient $\nabla F_i(x_k)$ is equal and

*Trivially, the triangle inequality implies $\|\nabla F(x_k) - g_k\| \leq \sum_{i=1}^{N}\frac{1}{N}\|\omega_i\|$. Then, by using Jensen's inequality we obtain $\|\nabla F(x_k) - g_k\|^2 \leq \left(\sum_{i=1}^{N}\frac{1}{N}\|\omega_i\|\right)^2 \leq \frac{1}{N}\sum_{i=1}^{N}\|\omega_i\|^2$. As a consequence, $\mathbb{E}_k[\|\nabla F(x_k) - g_k\|^2] \leq \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_k[\|\omega_i\|^2] = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n}\mathbb{E}_k[(w_i)_j^2] \leq n\sigma$, and Assumption 2.4 holds.

bounded by $M > 0$, then the variance of g_k is bounded by $\frac{M}{|S_k|}$ (see [9, §5.2]). Assumption 2.4 implies

$$\mathbb{E}_k[\|g_k\|^2] \leq M_1 + M_2\|\nabla F(x_k)\|^2, \quad \forall\ k \geq 0, \tag{2.15}$$

with $M_1 = M_g$ and $M_2 = 1$. Inequality (2.15) is generally assumed in the analysis of SG methods and also in the analysis of the first-order version of TRish presented in [15]; clearly, it is a milder restriction on the variance of g_k than Assumption 2.4.

THEOREM 2.2. *Let Assumptions 2.1, 2.3 and 2.4 hold, β as in (2.9), and suppose*

$$\alpha < \frac{\gamma_2}{2\gamma_1^2 L}. \tag{2.16}$$

1. *If Assumption 2.2 holds and $\alpha < \min\left\{\frac{\gamma_2}{2\gamma_1^2 L}, \frac{1}{\mu\gamma_2}\right\}$ then*

$$\mathbb{E}[F(x_{k+1})] - F_* \overset{k\to\infty}{\longrightarrow} \frac{2\beta M_g}{\mu\gamma_2}.$$

2. *If f is a general function*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(x_k)\|^2\right] \overset{K\to\infty}{\longrightarrow} \frac{4\beta M_g}{\gamma_2}.$$

Proof. 1. By subtracting F_* on both sides in (2.11) and using Assumption 2.4, we obtain

$$\mathbb{E}_k[F(x_{k+1})] - F_* \leq F(x_k) - F_* - \frac{1}{2}\alpha\left(\gamma_2 - \alpha\gamma_1^2 L\right)\|\nabla F(x_k)\|^2 + \alpha\beta M_g.$$

Due to the bound on α, we have $\gamma_2 - \alpha\gamma_1^2 L > \gamma_2/2 > 0$, and applying the PL condition (2.3) yields

$$\mathbb{E}_k[F(x_{k+1})] - F_* \leq \left(1 - \frac{1}{2}\alpha\mu\gamma_2\right)(F(x_k) - F_*) + \alpha\beta M_g. \tag{2.17}$$

By taking total expectations, letting $\xi = (1 - \frac{1}{2}\alpha\mu\gamma_2)$ and proceeding by induction, we obtain

$$\begin{aligned}
\mathbb{E}[F(x_{k+1})] - F_* &\leq \xi(\mathbb{E}[F(x_k)] - F_*) + \alpha\beta M_g \\
&\leq \xi(\xi(\mathbb{E}[F(x_{k-1})] - F_*) + \alpha\beta M_g) + \alpha\beta M_g \\
&\leq \cdots \leq \xi^{k+1}(\mathbb{E}[F(x_0)] - F_*) + \sum_{j=0}^{k}\xi^k\alpha\beta M_g.
\end{aligned}$$

The upper bound on α and the relations $\mu \leq L$, $\gamma_2 < \gamma_1$, imply $\xi \in (0,1)$ and $\sum_{j=0}^{\infty}\xi^k = \frac{1}{1-\xi}$. The claim follows by noting that $\frac{\alpha}{1-\xi} = \frac{2}{\gamma_2\mu}$.

2. By applying Assumption 2.4, inequality (2.11) becomes

$$\mathbb{E}_k[F(x_{k+1})] \leq F(x_k) - \frac{1}{4}\alpha\gamma_2\|\nabla F(x_k)\|^2 + \alpha\beta M_g.$$

By taking the total expectation, we get, for all k,

$$\mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{4}{\alpha\gamma_2}(\mathbb{E}[F(x_k)] - \mathbb{E}[F(x_{k+1})]) + \frac{4\beta M_g}{\gamma_2}.$$

By summing this inequality for $k = 1, \ldots, K$ and using that F is bounded below by F_*, we obtain

$$\mathbb{E}\left[\sum_{k=1}^{K}\|\nabla F(x_k)\|^2\right] \leq \frac{4}{\alpha\gamma_2}(F(x_1) - F_*) + K\frac{4\beta M_g}{\gamma_2},$$

and the claim follows. □

Theorem 2.2 holds for any value $\gamma_1/\gamma_2 > 1$ and differs in this respect from [15, Theorem 1, Theorem 5], which prove, under condition (2.15), results on both the expected decrease of the optimality gap and the expected decrease of the average square norm of the gradient. In fact, a restriction on the parameters γ_1, γ_2 of the form $\gamma_1 - h_2(\gamma_1 - \gamma_2) > 0$, with $h_2 \geq \frac{1}{2}\sqrt{M_1} + \sqrt{M_2}$, is required in [15]. Such restriction on the choice of γ_1, γ_2 is irrelevant if $h_2 \approx 1$, whereas it becomes stringent as h_2 increases. The need to assume a constraint on γ_1/γ_2 in [15] depends on the more general assumption (2.15) imposed in place of Assumption 2.4.

We also observe that Theorem 2.2 improves upon [16, Theorem 4.1, Theorem 4.4], which are also based on Assumption 2.4, in the values of the asymptotic expected optimality gap and average square norm of the gradient. Specifically, the asymptotic expected optimality gap $\frac{2\beta M_g}{\mu\gamma_2}$ provided in Item 1 of Theorem 2.2 is smaller than $\left(\frac{\gamma_1^2}{\gamma_2^2} - \frac{1}{2}\right)\frac{M_g}{\mu}$, due to the definition of β in (2.9) and the restriction of α in (2.16), whereas the asymptotic expected optimality gap provided in [16, Theorem 4.4] takes values $4\left(\frac{\gamma_1^2}{\gamma_2^2} - \frac{1}{8}\right)\frac{M_g}{\mu}$. As for the asymptotic average square norm of the gradient, the value $\frac{4\beta M_g}{\gamma_2}$ provided in Item 2 of Theorem 2.2 is smaller than $\left(2\frac{\gamma_1^2}{\gamma_2^2} - 1\right)M_g$, due again to (2.9) and (2.16), whereas the asymptotic expected optimality gap provided in [16, Theorem 4.1] takes values $\left(8\frac{\gamma_1^2}{\gamma_2^2} - 1\right)M_g$.

Finally, we note that the asymptotic expected optimality gap and the average square norm of the gradient are proportional to β in (2.9), which decreases as both $\gamma_1 - \gamma_2$ and α decrease. On the other hand, the closer $\frac{\gamma_1}{\gamma_2}$ is to one, the closer the performance of TRish algorithm becomes to that of the SG algorithm; likewise, the smaller α, the slower the rate to achieve the limit value.

We conclude our analysis by considering the case where condition (2.15) holds with $M_1 = 0$ and for some positive M_2. Under this assumption, the optimality gap and the expected average square norm of the gradient eventually vanish. This case covers the implementation of TRish considered in the next sections.

THEOREM 2.3. *Let Assumptions 2.1 and 2.3 hold. Suppose that (2.15) holds with $M_1 = 0$ and that*

$$\left(\frac{\gamma_2}{\gamma_1}\right)^2 > 1 - \frac{1}{4M_2}, \qquad \alpha < \frac{1}{4\gamma_2 L M_2}.$$

1. *If Assumption 2.2 holds and $\alpha < \min\left\{\frac{1}{4\gamma_2 L M_2}, \frac{2\gamma_2}{\mu\gamma_1^2}\right\}$, then*

$$\mathbb{E}[F(x_{k+1})] - F_* \overset{k\to\infty}{\longrightarrow} 0.$$

2. *If f is a general function, then*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(x_k)\|^2\right] \overset{K\to\infty}{\longrightarrow} 0.$$

Proof. Recalling the definition (2.9) of β, by (2.10) we have

$$\begin{aligned}\mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \alpha\left(\frac{\gamma_1^2}{2\gamma_2} - \beta M_2\right)\|\nabla F(x_k)\|^2 \\ &\leq F(x_k) - \alpha\frac{\gamma_1^2}{4\gamma_2}\|\nabla F(x_k)\|^2, \end{aligned} \tag{2.18}$$

where the last inequality follows from $\beta < \frac{\gamma_1^2}{4\gamma_2 M_2}$ and $\frac{\gamma_1^2}{2\gamma_2} - \beta M_2 > \frac{\gamma_1^2}{4\gamma_2}$.

1. By subtracting F_* on both sides and using the PL condition (2.3), we obtain

$$\mathbb{E}_k[F(x_{k+1})] - F_* \leq \left(1 - 2\alpha\mu\frac{\gamma_1^2}{4\gamma_2}\right)(F(x_k) - F_*). \tag{2.19}$$

By taking total expectations, letting $\xi = 1 - \alpha\mu\frac{\gamma_1^2}{2\gamma_2}$ and proceeding by induction, we obtain

$$\mathbb{E}[F(x_{k+1})] - F_* \leq \xi(\mathbb{E}[F(x_k)] - F_*) \leq \ldots \leq \xi^{k+1}(\mathbb{E}[F(x_0)] - F_*). \tag{2.20}$$

Then, since $\xi \in (0,1)$ by definition, the claim follows.

2. Taking total expectations on (2.18) yields, for all k,

$$\mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{4\gamma_2}{\alpha\gamma_1^2}(\mathbb{E}[F(x_k)] - \mathbb{E}[F(x_{k+1})]).$$

By summing this inequality for $k = 1, \ldots, K$ and using that F is bounded below by F_*, we obtain

$$\mathbb{E}\left[\sum_{k=1}^{K} \|\nabla F(x_k)\|^2\right] \leq \frac{4\gamma_2}{\alpha\gamma_1^2}(F(x_1) - F_*),$$

and the claim follows. □

The claims of Theorem 2.3 are similar to [15, Theorem 1, Theorem 5] when one assumes $M_1 = 0$ in (2.15), and they draw to conclusions that are in accordance with the discussion in [15]. In particular, the restriction on $\frac{\gamma_1}{\gamma_2}$ in our theorem can be written as $\gamma_1^2 - 4M_2(\gamma_1^2 - \gamma_2^2) > 0$; such a condition slightly affects the choice of γ_1 and γ_2 if M_2 is moderate, whereas it becomes more restrictive for large values of M_2, since it implies $\gamma_1 \approx \gamma_2$ if $M_2 \gg 0$. This issue is observed also in [15, Theorem 1, Theorem 5], where the requirement on γ_1 and γ_2 is $\gamma_1 - \sqrt{M_2}(\gamma_1 - \gamma_2) > 0$.

The requirements of Theorem 2.15 and [15, Theorem 1, Theorem 5] on the hyper-parameters α, γ_1, γ_2, can be compared as follows. On the one hand, we observe that our upper bound on γ_1/γ_2 is more stringent than the one in [15]†. On the other hand, our bounds on the steplength α improve upon those given in [15, Theorem 1, Theorem 5] in the following respects. Specifically, suppose $\gamma_2 < \frac{1}{4}$ and $M_2 > \max\{\gamma_1, \frac{\gamma_1^2\mu}{8\gamma_2^2 L}\}$. First, note that the condition $M_2 > \frac{\gamma_1^2\mu}{8\gamma_2^2 L}$ is not restrictive since $\frac{\gamma_2}{\gamma_1} < 1$, $\mu \leq L$. Second, under Assumption 2.2 the result in [15, Eqn (17)] reads $\alpha \leq \min\left\{\alpha^*, \alpha^\dagger\right\} := \min\left\{\frac{1}{\mu(\gamma_1 - \sqrt{M_2}(\gamma_1 - \gamma_2))}, \frac{\gamma_1 - \sqrt{M_2}(\gamma_1 - \gamma_2)}{\gamma_1 L M_2}\right\}$ and the condition $\gamma_1 < M_2$ easily yields $\alpha^\dagger = \min\left\{\alpha^*, \alpha^\dagger\right\}$. Our convergence results are proved under the assumption $\alpha < \frac{1}{4\gamma_2 L M_2}$ and $\frac{1}{4\gamma_2 L M_2} > \alpha^\dagger$ is trivially satisfied whenever $\gamma_2 < \frac{1}{4}$ since $\frac{\gamma_1}{4\gamma_2} > \gamma_1$ while $(\gamma_1 - \sqrt{M_2}(\gamma_1 - \gamma_2)) < \gamma_1$. Summarizing, our upper bound on the steplength results larger than the one provided in [15]. For a general function, the discussion above still holds if $\gamma_2 < \frac{1}{4}$.

We conclude this section by summarizing our results. First, in Lemma 2.1 we provided results for the expected decrease $\mathbb{E}_k[F(x_{k+1})] - F(x_k)$ in a simplified way with respect to [15] and without invoking the probability of the event $\nabla F(x_k)^T g_k \geq 0$. Second, in Theorem 2.2 we provided asymptotic expected values under Assumption 2.4 that are smaller than those derived in [16]; we remark that Assumption 2.4 was made in [16] and is stronger than the assumption made in [15] which is standard in the analysis of Stochastic Gradient methods. Third, in Theorem 2.3 we provided asymptotic expected values under condition (2.15) with $M_1 = 0$, which is guaranteed by the implementation of TRish considered in the following section; note that Theorem 2.3 improves the upper bound on the stepsize α under reasonable conditions on the parameters involved.

3. TRish with adaptive sampling. In this section, we present a practical implementation of TRish in compliance with the theoretical assumptions of the previous section. Our implementation is based on the adaptive selection of the sample sizes for computing the stochastic estimators g_k firstly proposed in [7].

†The requirement on γ_1 and γ_2 in [15, Theorem 1, Theorem 5] is $\gamma_1 - \sqrt{M_2}(\gamma_1 - \gamma_2) > 0$, i.e., $\frac{\gamma_2}{\gamma_1} < 1 - \frac{1}{\sqrt{M_2}}$. It is easy to see that for $M_2 > 1$ the condition $\left(\frac{\gamma_2}{\gamma_1}\right)^2 > 1 - \frac{1}{4M_2}$ in Theorem 2.3 is more restritive.

3.1. Adaptive strategies for computing g_k. Building the approximate gradient g_k is a crucial and delicate issue in optimization methods based on random models. In our case, due to the form (1.1) of F, it is straightforward to consider a mini-batch gradient approximation. Thus, letting $S \subset \{1, \dots, N\}$ be a sample, we denote

$$\nabla F_S(x) = \frac{1}{|S|} \sum_{i \in S} \nabla F_i(x), \tag{3.1}$$

and we let g_k be of the form

$$g_k = \nabla F_{S_k}(x_k), \tag{3.2}$$

which satisfies Assumption 2.3 if S_k is uniformly chosen at random without replacement.

Forming g_k leaves open the choice of the cardinality of the set S_k, which should be chosen in such a way that the assumptions made on g_k in Section 2 hold. A series of works [1–4, 6–10, 12] present a dynamic choice of the sample size. In [3, 4], such a choice is ruled by the combination of the Trust-Region and the Inexact Restoration procedures. In [1, 2, 6, 10, 12], the sample size is ruled by the Chebyshev inequality or the Bernstein concentration inequality. In [7, 8], the authors propose a strategy for computing the sample size that enforces the theoretical properties specified in our analysis for g_k; in practice, they offer criteria for testing whether the current sample size should be maintained or increased. The common approach in [7, 8] is to estimate the variance in the stochastic gradient or in random variables depending on the stochastic gradient, and increase the sample size if necessary. The heuristic relies on estimating the variance by the sample variance; if such an estimation suggests an increase of the sample size, it is assumed that the increase is small enough to consider reliable estimating the variance of the new sample by the variance of the current one. Now we summarize the strategies developed in [7, 8].

Condition (2.12) with $M_1 = 0$ can be enforced by adopting the strategy proposed by Bollapragrada et al., in [7], where g_k is supposed to satisfy

$$\mathbb{E}_k \left[(g_k^T \nabla F(x_k) - \|\nabla F(x_k)\|^2)^2 \right] \leq \theta^2 \|\nabla F(x_k)\|^4, \quad \text{for some } \theta > 0, \tag{3.3}$$

and

$$\mathbb{E}_k \left[\left\| g_k - \frac{g_k^T \nabla F(x_k)}{\|\nabla F(x_k)\|^2} \nabla F(x_k) \right\|^2 \right] \leq \nu^2 \|\nabla F(x_k)\|^2, \quad \text{for some } \nu > 0. \tag{3.4}$$

Condition (3.3) represents an upper bound on the variance of $g_k^T \nabla F(x_k)$, while condition (3.4) represents an upper bound on the variance of the component of g_k that is orthogonal to the true gradient $\nabla F(x_k)$. Under these conditions, (2.15) holds with $M_1 = 0$ and $M_2 = 1 + \theta^2 + \nu^2$, see [7, Lemma 3.1]. Since the above conditions are too expensive to be implemented, the variances on the left-hand sides are approximated with the corresponding sample variances and the true gradient $\nabla F(x_k)$ with a sample gradient. Hence, the two conditions above are replaced as follows. Instead of condition (3.3), we use the approximate inner product test

$$\begin{aligned} &\mathrm{Var}_{i \in S_k} (\nabla F_i(x_k)^T g_k) = \frac{1}{|S_k| - 1} \sum_{i \in S_k} \left(\nabla F_i(x_k)^T g_k - \|g_k\|^2 \right)^2, \\ &\frac{\mathrm{Var}_{i \in S_k} (\nabla F_i(x_k)^T g_k)}{|S_k|} \leq \theta^2 \|g_k\|^4. \end{aligned} \tag{3.5}$$

Instead of condition (3.4), we use the approximate orthogonality test

$$\begin{aligned} &\mathrm{Var}_{i \in S_k} \left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T g_k}{\|g_k\|^2} g_k \right) = \frac{1}{|S_k| - 1} \sum_{i \in S_k} \left\| \nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T g_k}{\|g_k\|^2} g_k \right\|^2 \\ &\mathrm{Var}_{i \in S_k} \left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T g_k}{\|g_k\|^2} g_k \right) \leq \nu^2 \|g_k\|^2 \end{aligned} \tag{3.6}$$

If (3.5) and (3.6) are satisfied, the sample size is kept unchanged, otherwise it is enlarged [7, §4.3].

3.2. Algorithm implementation. Our implementation is called TRish_AS (TRish with Adaptive Sampling) and combines TRish with the strategy based on conditions (3.5) and (3.6). The resulting algorithm is in accordance with the requirements on g_k specified in our theoretical analysis. Furthermore, it offers an adaptive choice of g_k and thus a rule for selecting the parameters $\{|S_k|\}$.

We explain the main steps for computing the sample size in TRish_AS. If conditions (3.5) and (3.6) are satisfied, then the sample size is kept unchanged. Otherwise, by using the simplifying assumptions in [7, §4.3], a sample size larger than S_k is computed as

$$\min\left\{\max\left\{\left\lceil\frac{\mathrm{Var}_{i\in S_k}(\nabla F_i(x_k)^T g_k)}{\theta^2\|g_k\|^4}\right\rceil, \left\lceil\frac{\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k)-\frac{\nabla F_i(x_k)^T g_k}{\|g_k\|^2}g_k\right)}{\nu^2\|g_k\|^2}\right\rceil\right\}, N\right\}. \quad (3.7)$$

Finally, we use the sample control in the noisy regime fully described in [7, §4.2] and sketched at Step 5 of Algorithm 2. If the sample sizes do not change along r iterations, the average of the most r recent sample gradients $g_{\mathrm{avg}} = \frac{1}{r}\sum_{j=k-r+1}^{k}\nabla F_{S_j}(x_j)$ is computed. Then, if the norm of g_{avg} is small compared to the norm of g_k, (3.5) and (3.6) are tested using g_{avg} instead of $\nabla F_{S_k}(x_k)$; in case of failure of one of such conditions, the sample size is increased using (3.7) and g_{avg} instead of g_k. We report TRish_AS in the following Algorithm 2.

Algorithm 2 TRish_AS: practical implementation

Choose an initial iterate $x_0 \in \mathbb{R}^n$, a steplength $\alpha > 0$, a sample size $S_0 \in \{1,\dots,N\}$ an integer $r > 0$ and real parameters $0 < \gamma_2 < \gamma_1$, $\theta > 0$, $\nu > 0$.

Set $k = 0$. Compute $g_0 = \nabla F_{S_0}(x_0)$.
Repeat until a convergence test is satisfied

1. Compute

$$x_{k+1} = x_k - \begin{cases}\gamma_1\alpha g_k, & \text{if } \|g_k\| \in [0, \frac{1}{\gamma_1})\\ \alpha\dfrac{g_k}{\|g_k\|}, & \text{if } \|g_k\| \in [\frac{1}{\gamma_1}, \frac{1}{\gamma_2}]\\ \gamma_2\alpha g_k, & \text{if } \|g_k\| \in (\frac{1}{\gamma_2}, +\infty[.\end{cases} \quad (3.8)$$

2. Set $k = k + 1$.
3. Set $|S_k| = |S_{k-1}|$, choose a new sample S_k and form $g_k = \nabla F_{S_k}(x_k)$.
4. If (3.5) or (3.6) is not satisfied then
   update $|S_k|$ using (3.7), choose a new sample S_k and form $g_k = \nabla F_{S_k}(x_k)$.
5. If $|S_k| = |S_{k-1}| = \cdots = |S_{k-r}|$ then
   compute $g_{\mathrm{avg}} = \frac{1}{r}\sum_{j=k-r+1}^{k}\nabla F_{S_j}(x_j)$.
   If $\|g_{\mathrm{avg}}\| < \gamma\|\nabla F_{S_k}(x_k)\|$ then
      If (3.5) or (3.6) is not satisfied using g_{avg} instead of $\nabla F_{S_k}(x_k)$ then
         update $|S_k|$ using (3.7) and g_{avg} instead of g_k;
         choose a new sample S_k and form $g_k = \nabla F_{S_k}(x_k)$.

4. Numerical experience. We now present the numerical performance of Algorithm 2. Our goal is to show that the overall performance of TRish may benefit from this adaptive approach, instead of using a prefixed and constant sample size as done in [15, 16].

4.1. Logistic Regression. We consider datasets with feature vectors $z_i \in \mathbb{R}^n$, labels $y_i \in \{-1, 1\}$, $i = 1,\dots,N$, and address the problem of binary classification by minimizing the logistic regression function

$$F(x) = \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + e^{-y_i(x^T z_i)}\right).$$

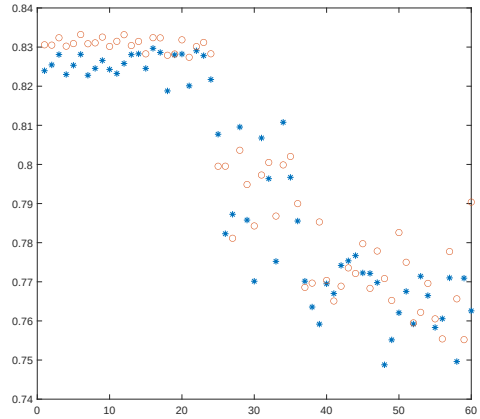

FIG. 4.1. *Dataset a1a. Average testing accuracy at termination. TRish: symbol "*", TRish_AS: symbol "o".*

The datasets considered are available in the LIBSVM repository [11]. A testing dataset $\{(\bar{z}_i, \bar{y}_i)\}_{i=1}^{\bar{N}}$ is used for evaluating the accuracy in classifying unseen data.

Our experiments were carried out as in [15]. We considered 60 parameter settings $(\alpha, \gamma_1, \gamma_2)$ defined as follows. The stepsize takes value $\alpha \in \{0.1, 10^{-\frac{1}{2}}, 1, 10^{\frac{1}{2}}, 10\}$. The parameters γ_1, γ_2 take value $\gamma_1 \in \{\frac{4}{G}, \frac{8}{G}, \frac{16}{G}, \frac{32}{G}\}$, $\gamma_2 \in \{\frac{1}{2G}, \frac{1}{G}, \frac{2}{G}\}$ with G being the average norm of stochastic gradients generated by SG run for one epoch with stepsize $\alpha = 0.1$ and sample size equal to 64. A complete set of experiments amounts to applying both TRish and TRish_AS with the 60 different combinations $(\alpha, \gamma_1, \gamma_2)$. Both algorithms were run for one epoch. TRish was run using a sample size equal to 64, see [15], while the initial sample size for TRish_AS was $S_0 = \min\{32, \lceil \frac{N}{100} \rceil\}$. Regarding the implementation of condition (3.7) in TRish_AS, we fixed $\theta = 0.9$ and $\nu = 5.84$, due to their statistical significance (see [7, §4.4]) and $r = 10$. In a few cases, the evaluation of (3.7) in finite precision was harmful, due to very large or small norm of g_k, and affected by underflow or overflow; in such occurrences the sample size was not updated.

In the remainder of the section, we report the following results. For each parameter setting $(\alpha, \gamma_1, \gamma_2)$ we measure, averaging over 50 runs, the testing accuracy defined as the fraction of the testing set correctly classified and, limited to TRish_AS, the sample size reached at termination. Then, for each test problem and parameter setting we plot the average testing accuracy reached by TRish (star symbol) and TRish_AS (circle symbol). For the sake of clarity, we also provide the same plots separately for each fixed value of α and all tested values of γ_1, γ_2, i.e., we focus on the 12 parameter settings sharing the same value of α. For each test problem we also present the plots on the average training loss and testing accuracy versus the average number of effective gradient evaluations (EGE), that is the number of full gradient evaluations. Finally, in a table we summarize the statistics corresponding to the best run for TRish (first row), i.e., the case where TRish reached the best average accuracy, and the statistics corresponding to the best run for TRish_AS (second row). On each row, we report the parameter setting $(\alpha, \gamma_1, \gamma_2)$ that provided the best run and, for such a triplet, the average testing accuracy obtained by the two methods and the average final sample size used by TRish_AS; thus, it is possible to compare the results obtained by the two algorithms with the same parameter setting. Correspondingly to such runs we also give statistics on the steps taken by TRish and TRish_AS among the normalized step and the SG-type step (Cases 1–3).

4.1.1. a1a. We used a training set consisting of $N = 1605$ points, features vectors have dimension $n = 123$. The testing set consists of $\bar{N} = 29351$ points. We performed the runs using the value $G = 0.3477$ for the setting of γ_1 and γ_2. In Figure 4.1 we plot the average testing accuracy reached by TRish and by TRish_AS, while in Figure 4.2 each plot displays the average testing accuracies reached for specific value of α and all tested values of γ_1, γ_2.

The results in Figure 4.1 show that the accuracy attained with TRish_AS is higher than the accuracy obtained by TRish in 46 cases out of 60. The minimum and maximum averaged sample

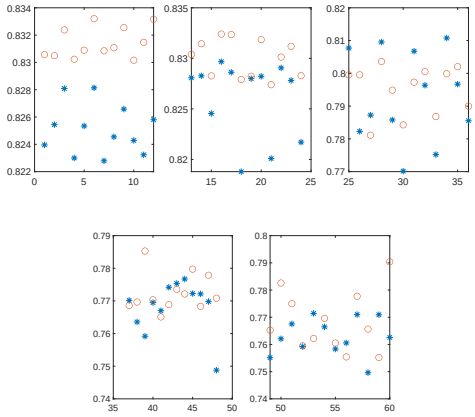

FIG. 4.2. *Dataset a1a. Average testing accuracy at termination. Runs with fixed steplength α and varying γ_1, γ_2. Top, from left to right: $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. Bottom, from left to right: $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. TRish: symbol "*", TRish_AS: symbol "o".*

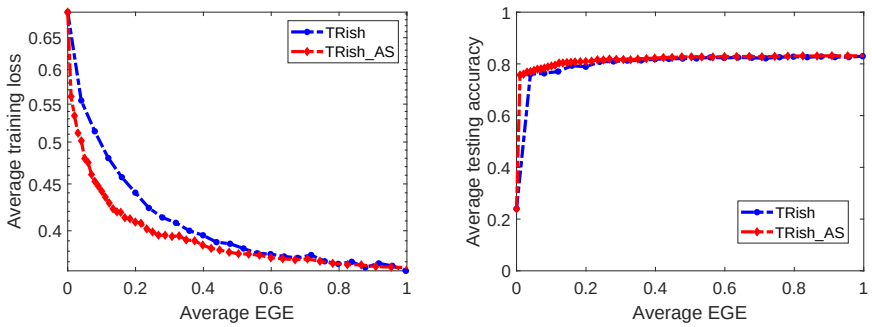

FIG. 4.3. *Dataset a1a. Average training loss (left) and testing accuracy (right) versus one epoch for TRish (blue) and TRish_AS (red).*

sizes at termination of TRish_AS were 60 and 186 respectively. We also observe in Figure 4.2 that when the steplengths $\alpha = 10^{-1}, 10^{-\frac{1}{2}}$ are used, TRish_AS is quite insensitive to the choice of γ_1 and γ_2 and TRish_AS is more accurate than TRish in classification. In Figures 4.3 we show the behaviour of the average training loss and testing accuracy versus the average number of effective gradient evaluations (EGE) for the two methods equipped with their own best parameters settings.

Table 4.1 concerns the best (averaged) run and shows that TRish_AS outperforms TRish for its best parameters setting, as well as for the one which provides the best averaged run of TRish. Moreover, the average final sample sizes of TRish_AS are considerably small, as they represent the 7.7% of the whole sample for the first parameter combination, and the 4.7% for the second parameter combination, respectively. We also give statistics regarding the step taken along the iterations among the normalized step (Case 2) and the SG-type steps (Cases 1 and 3). Focusing on the runs with parameters given in the second row of Table 4.1, on average TRish selected the normalized step (Case 2) 33% times and the SG-type step (Case 3) 67% times, while on average TRish_AS selected the normalized step (Case 2) 19% times and the SG-type step (Case 3) 81% times.

4.1.2. w1a. This training data set consists of $N = 2477$ points, features vectors have dimension $n = 300$. The testing set consists of $\bar{N} = 47272$ points. Following [15] we performed the runs using the value $G = 0.0887$ for the setting of γ_1 and γ_2.

Figure 4.4 shows that TRish_AS obtained the highest accuracy in 47 cases out of 60. The minimum and maximum average sample sizes of TRish_AS at termination were 67 and 343 respectively. Figure 4.5 clarifies that the accuracies achieved by the two methods are similar, and

	$(\alpha, \gamma_1, \gamma_2)$	Testing accuracy		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	$(0.3162, 23.9593, 1.4975)$	0.8297	0.8324	124
Best run of TRish_AS	$(0.1000, 23.9593, 5.9898)$	0.8281	0.8332	75

TABLE 4.1

Dataset a1a. First row: parameter setting corresponding to the best run for TRish in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.

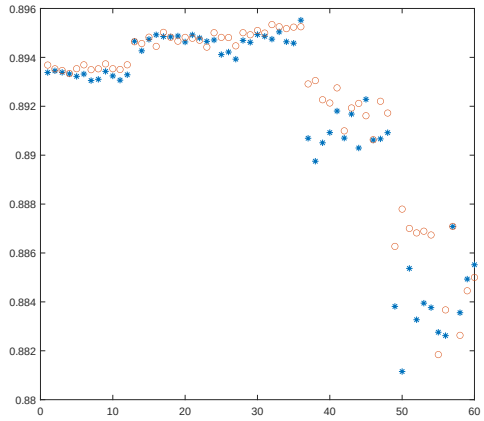

FIG. 4.4. *Dataset w1a. Average testing accuracy at termination. TRish: symbol "*", TRish_AS: symbol "o".*

that the performance of TRish_AS is only slightly affected by the choice of parameters γ_1 and γ_2. In Figures 4.6 we show the behaviour of the average training loss and testing accuracy versus the average number of EGE for the methods equipped with their own best parameters settings. We observe that TRish reaches smaller values of the training loss, while the two methods achieves almost the same accuracy.

The good behaviour of TRish_AS is confirmed in Table 4.2 as TRish_AS is comparable to TRish in the best performance of the latter. In the reported runs the sample size at termination is around the 7.4% and 9.8% of N respectively. When the parameters were selected as in the second row of Table 4.2, TRish computed the normalized step (Case 2) 85% times and the SG-type step (Case 1 and 3) 15% times on average, whereas TRish_AS computed the normalized step (Case 2) 83% times and the SG-type step (Case 1 and 3) 17% times on average.

	$(\alpha, \gamma_1, \gamma_2)$	Testing accuracy		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	$(1.0000, 105.8551, 6.6159)$	0.8955	0.8953	183
Best run of TRish_AS	$(1.0000, 52.9276, 3.3080)$	0.8947	0.8953	242

TABLE 4.2

Dataset w1a. First row: parameter setting corresponding to the best run for TRish in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.

4.1.3. rcv1. This training data set consists of $N = 20242$ points, features vectors have dimension $n = 47236$. The testing set consists of $\bar{N} = 677399$ points. Following [15] we performed the runs using the value $G = 0.0497$ for the setting of γ_1 and γ_2. Figure 4.7 shows that TRish_AS outperforms TRish for almost all parameters combinations; more specifically, TRIsh_AS is more accurate than TRish 59 out of 60 runs. However, both methods perform quite well and the lowest

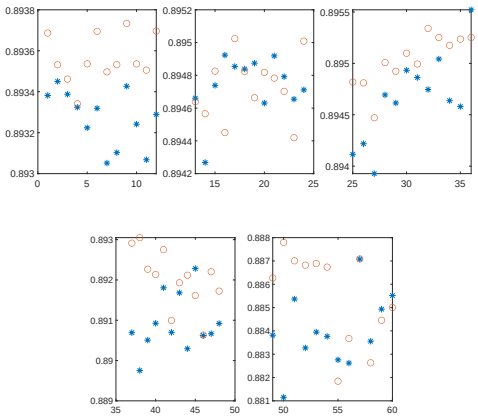

FIG. 4.5. *Dataset w1a. Average testing accuracy at termination. Runs with fixed steplength α and varying γ_1, γ_2. Top, from left to right: $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. Bottom, from left to right: $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. TRish: symbol "*", TRish_AS: symbol "o".*

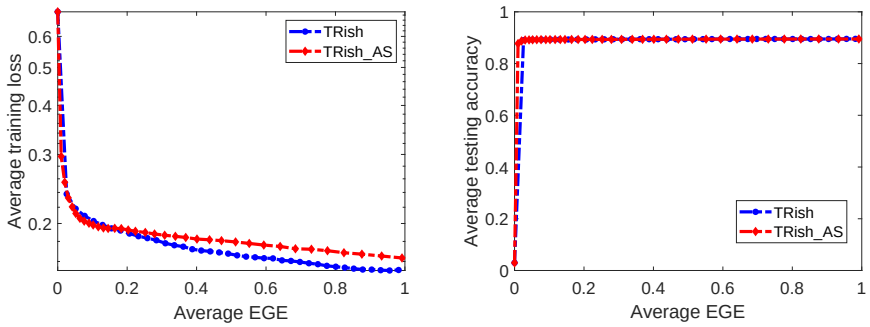

FIG. 4.6. *Dataset w1a. Average training loss (left) and testing accuracy (right) versus one epoch for TRish (blue) and TRish_AS (red).*

fraction of correctly classified features is approximately 0.9303 attained by TRish with $\alpha = 10^{-1}$, see Figure 4.8. Figure 4.9 shows the decrease of the average training loss and testing accuracy versus the average number of effective gradient evaluations for the methods equipped with their own best parameters settings. The best run of each procedure correctly classified 95% of instances in the testing set as reported in Table 4.3. The minimum averaged sample size at termination of TRish_AS was 32, while the maximum was 178 corresponding to 0.87% of N. Finally, considering the parameters setting at the second row of Table 4.3, we note that TRish selected the normalized step (Case 2) 75% times and the SG-type step (Case 1 and 3) 25% times on average, whereas TRish_AS selected the normalized step (Case 2) 69% times and the SG-type step (Case 3) 31% times on average.

	$(\alpha, \gamma_1, \gamma_2)$	Testing accuracy		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	(3.1623, 121.3961, 30.3490)	0.9581	0.9580	116
Best run of TRish_AS	(3.1623, 60.6980, 15.1745)	0.9579	0.9582	109

TABLE 4.3

Dataset rcv1. First row: parameter setting corresponding to the best run for TRish in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.

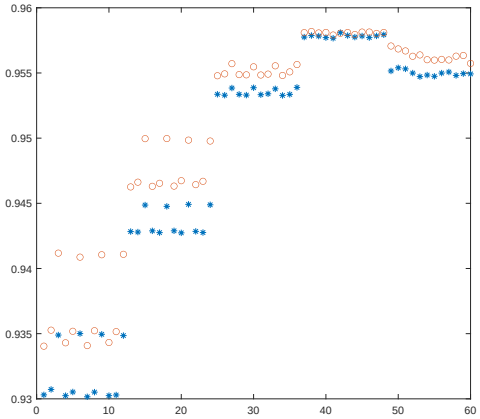

FIG. 4.7. *Dataset rcv1. Average testing accuracy at termination. TRish: symbol "*", TRish_AS: symbol "o".*

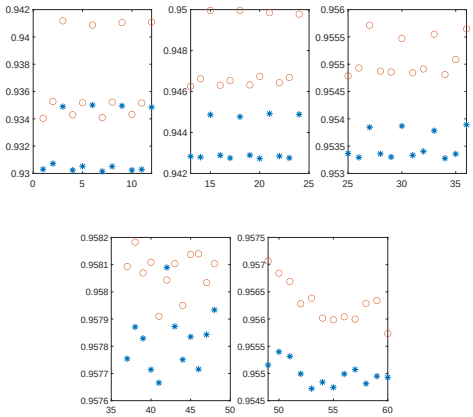

FIG. 4.8. *Dataset rcv1. Average testing accuracy at termination. Runs with fixed steplength α and varying γ_1, γ_2. Top, from left to right: $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. Bottom, from left to right: $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. TRish: symbol "*", TRish_AS: symbol "o".*

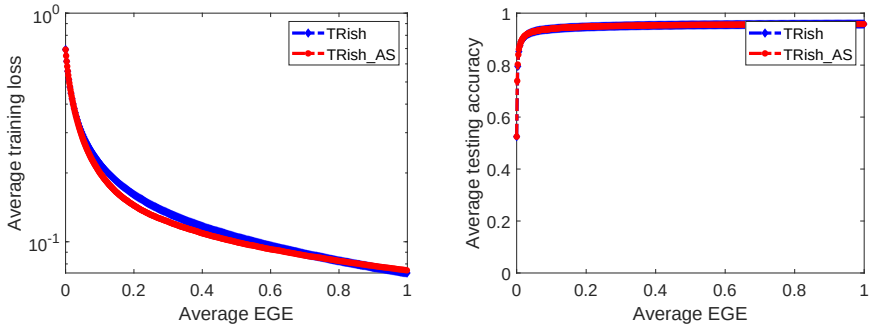

FIG. 4.9. *Dataset rcv1. Average training loss (left) and testing accuracy (right) versus one epoch for TRish (blue) and TRish_AS (red).*

4.1.4. Cina. This training data set consists of $N = 10000$ points, features vectors have dimension $n = 132$. The testing set consists of $\bar{N} = 6033$ points. The dataset was normalized such that each feature vector has components in the interval $[0, 1]$. We performed the runs using the value $G = 0.2517$ for the setting of γ_1 and γ_2.

TRish_AS achieves the highest accuracy in classification in 17 cases out of 60, see Figures 4.10 and 4.11. The accuracy is very similar for the two smallest steps, $\alpha = 10^{-1}$ and $\alpha = 10^{-\frac{1}{2}}$,

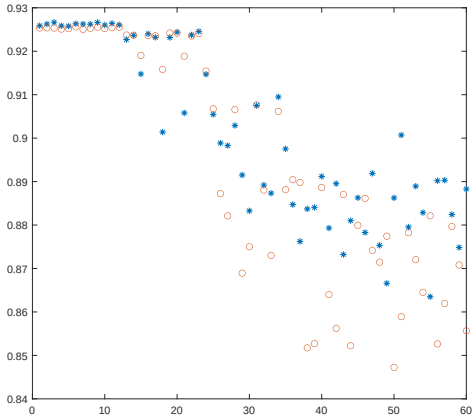

FIG. 4.10. *Dataset Cina. Average testing accuracy at termination. TRish: symbol "*", TRish_AS: symbol "o".*

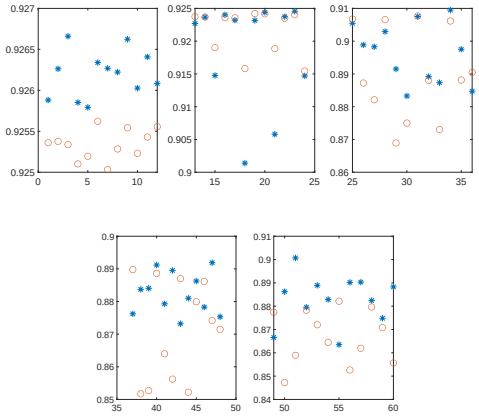

FIG. 4.11. *Dataset Cina. Average testing accuracy at termination. Runs with fixed steplength α and varying γ_1, γ_2. Top, from left to right: $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. Bottom, from left to right: $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. TRish: symbol "*", TRish_AS: symbol "o".*

which produce the highest accuracy. Overall the largest gain in accuracy of TRish with respect to TRish_AS is within 4.6%. The minimum and maximum averaged final values of the sample size of TRish_AS at termination were 409 and 771, respectively. In Figure 4.12 we show the behaviour of the average training loss and testing accuracy versus the average number of EGE for the methods equipped with their own best parameters settings.

The statistics corresponding to the best runs for both methods are reported in Table 4.4. In the reported runs the sample size of TRish_AS at termination is approximately 4.4% of N. Concerning the step taken, consider the second row of Table 4.4; on average TRish selected the normalized step (Case 2) 34% times and the SG-type step (Case 1 and 3) 66% times; on average TRish_AS selected the normalized step (Case 2) 25% times and the SG-type step (Case 3) 75% times.

4.2. Neural network based classification. In this second test, we address the problem of training a feedforward neural network for image classification. In particular, we consider the popular mnist dataset [19], which consists of a training set of $N = 60000$ binary images of handwritten digits from zero to nine, and a testing set of size $\bar{N} = 10000$. Our task is classifying whether an image of the testing set represents the digit "two" or not. We let $\{(z_i, y_i)\}_{i=1}^N$ be the training set, where $z_i \in \mathbb{R}^\ell$ is the binary image with $\ell = 784$ and $y_i \in \{0, 1\}$ is the corresponding label, where $y_i = 1$ detects the presence of a "two" and $y_i = 0$ its absence. We also denote with $h(\cdot; x) : \mathbb{R}^\ell \to \mathbb{R}$ the chosen feedforward neural network, where $x \in \mathbb{R}^n$ is the vector of all weights

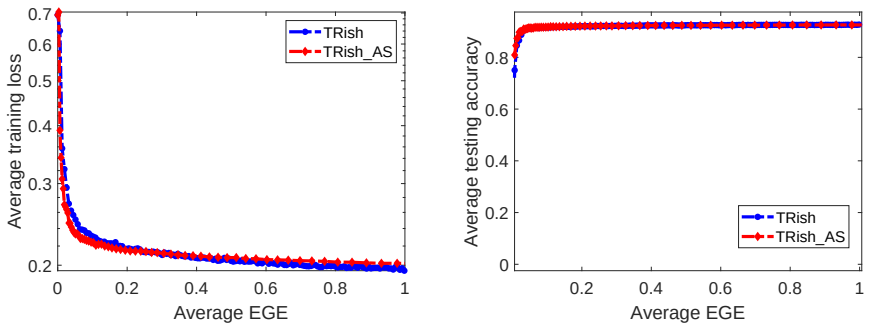

FIG. 4.12. *Dataset Cina. Average training loss (left) and testing accuracy (right) versus one epoch for TRish (blue) and TRish_AS (red).*

	$(\alpha, \gamma_1, \gamma_2)$	Testing accuracy		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	(0.1000, 18.4672, 9.2336)	0.9267	0.9253	448
Best run of TRish_AS	$(0.1000, 36.9344, 9.2336)$	0.9263	0.9256	436

TABLE 4.4

Dataset Cina. First row: parameter setting corresponding to the best run for TRish in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.

and biases. Note that the neural network has a single hidden layer equipped with $n_1 = 5$ nodes, whereas both the hidden and output layers are combined with the sigmoid activation function. As a result, the neural network possesses $n = (\ell + 2)n_1 + 1 = 3931$ parameters, which are computed by approximately solving the finite-sum minimization problem

$$\min_{x \in \mathbb{R}^n} F(x) \equiv -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log(h(z_i; x)) + (1 - y_i) \log(1 - h(z_i; x)) \right). \tag{4.1}$$

where the neural network is combined with the cross-entropy loss function. We performed a similar set of experiments as in Section 4.1, namely, we run both TRish and TRish_AS for 60 different parameter settings $(\alpha, \gamma_1, \gamma_2)$, where $\alpha \in \{0.1, 10^{\frac{1}{2}}, 1, 10^{\frac{1}{2}}, 10\}$, $\gamma_1 \in \{\frac{4}{G}, \frac{8}{G}, \frac{16}{G}, \frac{32}{G}\}$, $\gamma_2 \in \{\frac{1}{2G}, \frac{1}{G}, \frac{2}{G}\}$, being $G = 0.2091$ the average norm of the stochastic gradients generated by running SG for one epoch with stepsize $\alpha = 0.1$ and sample size equal to 64. For each combination of parameters, both algorithms are run for one epoch, by using a fixed sample size $S = 64$ for TRish, and an initial sample size $S_0 = \min\left\{32, \lceil \frac{N}{100} \rceil\right\}$ for TRish_AS.

As can be seen from Figure 4.13 and 4.14, TRish_AS is more accurate than TRish in 35 cases out of 60; furthermore we note that the accuracy achieved by TRish_AS is considerably higher than the accuracy obtained with TRish in several cases. Table 4.5 shows the highest testing accuracy achieved when the two algorithms are equipped with their own best parameters configuration; it can be seen that TRish_AS reached the largest testing accuracy among all the possible configurations. The minimum and maximum averaged sample sizes of TRish_AS at termination are 77 and 29887, respectively. Figure 4.15 shows the decrease of the average training loss and testing accuracy versus the average number of EGE for both methods equipped with their own best parameters settings. We can observe that the training loss in TRish_AS shows a steady behaviour as the computational cost increases, as opposed to the quite erratic behaviour of TRish. As for the steps taken by TRish and TRish_AS, we report once again the statistics related to the case where both methods are equipped with the best parameters configuration for TRish_AS (see second row of Table 4.5). On average, TRish selected the normalized step (Case 2) 10% times and the SG-type step (Case 1 and 3) 90% times, whereas TRish_AS selected on average the normalized

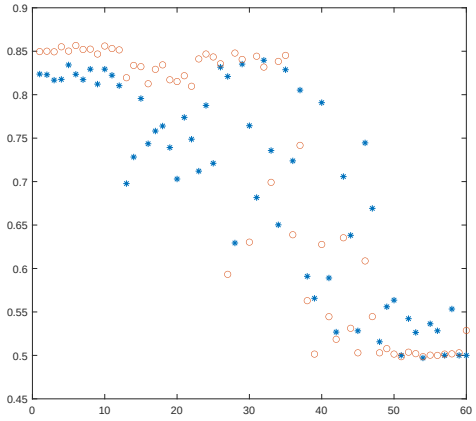

FIG. 4.13. *Dataset mnist. Average testing accuracy at termination. TRish: symbol "*", TRish_AS: symbol "o".*

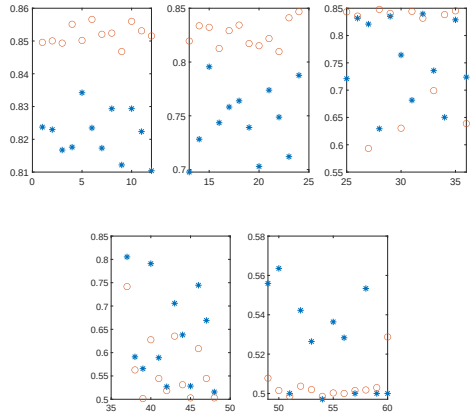

FIG. 4.14. *Dataset mnist. Average testing accuracy at termination. Runs with fixed steplength α and varying γ_1, γ_2. Top, from left to right: $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. Bottom, from left to right: $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. TRish: symbol "*", TRish_AS: symbol "o".*

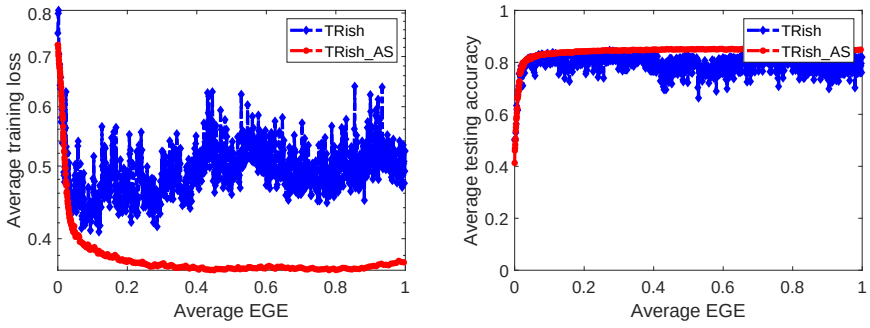

FIG. 4.15. *Dataset mnist. Average training loss (left) and testing accuracy (right) versus one epoch for TRish (blue) and TRish_AS (red).*

step (Case 2) 43% times and the SG-type step (Case 1 and 3) 57% times.

4.3. Neural network based regression. We now turn our attention to an optimization problem coming from nonlinear regression. Let $\{(z_i, y_i)\}_{i=1}^N$ be the training set, where $z_i \in \mathbb{R}^\ell$ represents the i-th feature vector, and $y_i \in \mathbb{R}$ is the corresponding target variable. By denoting with $h(\cdot; x) : \mathbb{R}^\ell \to \mathbb{R}$ a prediction function, we aim at estimating the parameters vector $x \in \mathbb{R}^n$

	$(\alpha, \gamma_1, \gamma_2)$	Testing accuracy		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	$(1.0000, 76.5184, 4.7824)$	0.8396	0.8318	1044
Best run of TRish_AS	$(0.1000, 38.2592, 9.5648)$	0.8235	0.8566	520

TABLE 4.5

Dataset mnist. First row: parameter setting corresponding to the best run for TRish in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing accuracy, average testing accuracy of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.

by addressing an optimization problem of the form (4.1).

In our tests, we use the AIR dataset employed in other previous works [4,17], which is available at the website [20]. This dataset 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. More specifically, each instance of the dataset consists of 8 features, including the concentrations of benzene, carbon monoxide, nitrogen oxides, ozone, non-metanic hydrocarbons, nitrogen dioxide, air temperature, and relative air humidity.

We aim at predicting the benzene concentration at a certain hour from the knowledge of the other $\ell = 7$ features. After removing the instances in which the benzene concentration is missing, reducing the dataset dimension from 9357 to 8991, we consider 70% of the dataset for training ($N = 6294$), and the remaining 30% for testing ($N_T = 2697$). In other words, the training phase is based on the data measured in the first 9 months, where the testing phase is performed using the data of the last 3 months. By letting $D = (d_{ij}) \in \mathbb{R}^{(N+N_T)\times\ell}$ be the matrix containing all the dataset examples along its rows, and defining the values $m_j = \min_{i=1,\ldots,N+N_T} d_{ij}$ and $M_j = \max_{i=1,\ldots,N+N_T} d_{ij}$ for $j = 1, \ldots, n$, we preprocess the data as below

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

so that all data values are constrained into the interval $[0, 1]$.

The prediction function $h(\cdot; x)$ is chosen as a feed-forward neural network with two hidden layers of 7 and 5 nodes respectively, as done in [17]. Consequently, $x \in \mathbb{R}^{102}$ represents the vector of weights and biases of the considered neural network. Each hidden layer is equipped with the linear activation function, whereas the output layer is combined with the sigmoid activation function.

We considered the same 60 parameter settings $(\alpha, \gamma_1, \gamma_2)$ as the ones defined in the previous sections, namely, $\alpha \in \{0.1, 10^{\frac{1}{2}}, 1, 10^{\frac{1}{2}}, 10\}$, $\gamma_1 \in \{\frac{4}{G}, \frac{8}{G}, \frac{16}{G}, \frac{32}{G}\}$, $\gamma_2 \in \{\frac{1}{2G}, \frac{1}{G}, \frac{2}{G}\}$, where $G = 0.0318$ is again the average norm of stochastic gradients generated by running SG for one epoch with stepsize $\alpha = 0.1$ and sample size equal to 64. For each combination of parameters, we run both TRish and TRish_AS for one epoch, by using a fixed sample size $S = 64$ for TRish, and an initial sample size $S_0 = \min\left\{32, \left\lceil \frac{N}{100} \right\rceil\right\}$ for TRish_AS.

As this is not a classification problem, we replace the testing accuracy with the evaluation of the testing loss at iteration k, i.e.,

$$F_{S_T}(x_k) = \frac{1}{N_T} \sum_{i \in S_T} \left(y_i - h(z_i; x_k)\right)^2,$$

where N_T is the dimension of the testing set, and $S_T \subseteq \{1, \ldots, N\}$, $|S_T| = N_T$, is the set of indexes corresponding to the instances of the testing set. Note that a lower value of the testing loss corresponds to a more accurate prediction of the target variable on the testing set.

In Figure 4.16, we report the average testing loss over 50 runs at termination for each parameters setting. We can appreciate once more the good performance of TRish_AS, which outperforms TRish for all 60 runs. The computational effort of TRish_AS is overall limited, as the minimum

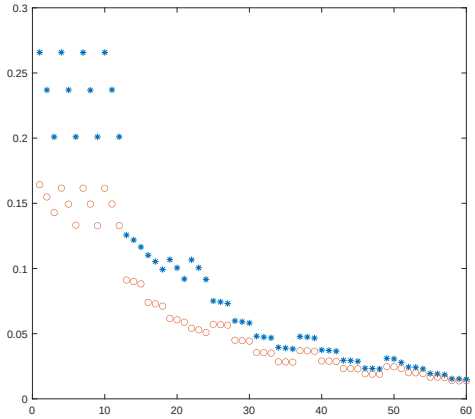

FIG. 4.16. *Dataset* AIR. *Average testing loss at termination. TRish: symbol "*", TRish_AS: symbol "o".*

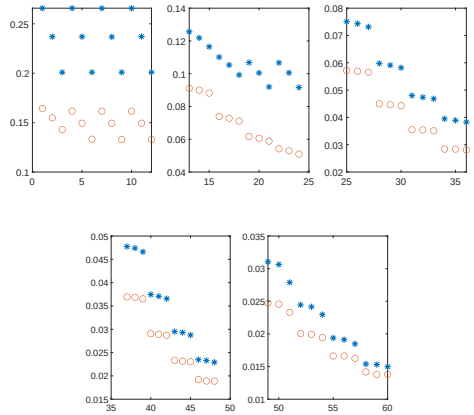

FIG. 4.17. *Dataset* AIR. *Average testing loss at termination. Runs with fixed steplength* α *and varying* γ_1, γ_2. *Top, from left to right:* $\alpha = 10^{-1}$, $\alpha = 10^{-\frac{1}{2}}$, $\alpha = 1$. *Bottom, from left to right:* $\alpha = 10^{\frac{1}{2}}$, $\alpha = 10$. *TRish: symbol "*", TRish_AS: symbol "o".*

and maximum averaged sample size at termination are 32 and 91, respectively. Figure 4.17 shows that TRish_AS is completely insensitive to the choice of γ_1 and γ_2 when $\alpha = 10^{-1}$, whereas TRish is always sensitive to these parameters for any choice of α. From Table 4.6, we see TRish_AS performs better than TRish even when it is equipped with the best parameters configuration for TRish. In Figure 4.18 we show the behaviour of the average training and testing loss versus the average number of effective gradient evaluations (EGE) for the methods equipped with their own best parameters settings. We can appreciate an accelerated behaviour of TRish_AS with respect to TRish in both training and testing loss. The statistics corresponding to the best runs for both methods are reported in Table 4.6. Concerning the step taken, consider the second row of Table 4.4; on average TRish selected the normalized step (Case 2) 81% times and the SG-type step (Case 3) 19% times; on average TRish_AS selected the normalized step (Case 2) 77% times and the SG-type step (Case 1 and 3) 23% times.

5. Conclusions. In this work a theoretical and experimental analysis of the stochastic trust-region based TRish algorithm was carried out. Our analysis complements and partly improves upon the existing results. Our implementation is based on a dynamic choice of the sample size, and provides good results with respect to using a prefixed and constant sample size. In particular, the performance of the algorithm is less sensitive to the choice of the parameters governing the form of the step, once the stochastic gradient is formed.

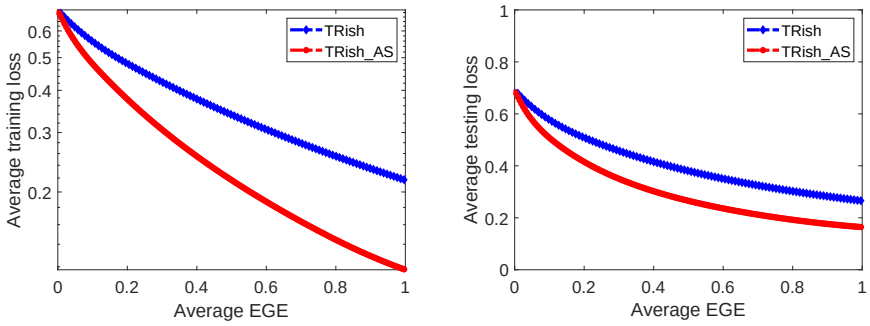

FIG. 4.18. *Dataset air. Average training loss (left) and testing loss (right) versus one epoch for TRish (blue) and TRish_AS (red).*

	$(\alpha, \gamma_1, \gamma_2)$	Testing loss		Average final $\lvert S_k \rvert$
		TRish	TRish_AS	TRish_AS
Best run of TRish	$(10, 1005.4492, 62.8406)$	0.01498	0.01382	91
Best run of TRish_AS	$(10, 1005.4492, 31.4203)$	0.01531	0.01379	83

TABLE 4.6

Dataset AIR. *First row: parameter setting corresponding to the best run for TRish in terms of testing loss, average testing loss of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet. Second row: parameter setting corresponding to the best run for TRish_AS in terms of testing loss, average testing loss of TRish and TRish_AS at termination with the triplet, average sample size of TRish_AS at termination with the triplet.*

REFERENCES

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

[2] S. Bellavia, G. Gurioli, B. Morini, Ph. L. Toint, *Trust-region algorithms: Probabilistic complexity and intrinsic noise with applications to subsampling techniques*, EURO Journal on Computational Optimization, 10, 100043, 2022.

[3] S. Bellavia, N. Krejić, B. Morini, *Inexact restoration with subsampled trust-region methods for finite-sum minimization*, Computational Optimization and Applications, 73, 701–736, 2020.

[4] S. Bellavia, N. Krejić, B. Morini, S. Rebegoldi, *A stochastic first-order trust-region method with inexact restoration for finite-sum minimization*, Computational Optimization and Applications, 84, 319–362, 2023.

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

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

[7] R. Bollapragada, R. Byrd, J. Nocedal, *Adaptive sampling strategies for stochastic optimization*, SIAM Journal on Optimization 28(4), 3312–3343, 2018.

[8] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, *Sample size selection in optimization methods for machine learning*, Mathematical Programming Ser. B, pp. 127-155, 2012.

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

[10] C. Cartis, K. Scheinberg, *Global convergence rate analysis of unconstrained optimization methods based on probabilistic models*, Mathematical Programming, 169, 337–375, 2018.

[11] C. C. Chang, C.J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2(3), 1–27, 2011.

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

[13] A. R. Conn, N.I.M. Gould, Ph.L. Toint, *Trust-region methods*, SMPS/SIAM Series on Optimization, 2000.

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

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

[16] F. E. Curtis, R. Shi, *A fully stochastic second-order trust region method*, Optimization Methods and Software, 37(3), 844-877, 2022.

[17] S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia, On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, Sensors and Actuators, B(129), 750–757,

2008.
[18] H. Karimi, J. Nutini, M. Schmidt, *Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 795–811, 2016.
[19] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11), 2278–2324, 1998.
[20] M. Lichman, UCI machine learning repository, https://archive.ics.uci.edu/ml/index.php, 2013.
[21] H. Robbins, S. Monro, *A Stochastic Approximation Method*, The Annals of Mathematical Statistics, 22 400–407, 1951.