A Subsampling Line-Search Method with Second-Order Results

E. Bergou * Y. Diouane† V. Kungurtsev ‡ C. W. Royer §

November 21, 2018

Abstract

In many contemporary optimization problems, such as hyperparameter tuning for deep learning architectures, it is computationally challenging or even infeasible to evaluate an entire function or its derivatives. This necessitates the use of stochastic algorithms that sample problem data, which can jeopardize the guarantees classically obtained through globalization techniques via a trust region or a line search. Using subsampled function values is particularly challenging for the latter strategy, that relies upon multiple evaluations. On top of that all, there has been an increasing interest for nonconvex formulations of data-related problems. For such instances, one aims at developing methods that converge to second-order stationary points, which is particularly delicate to ensure when one only accesses subsampled approximations of the objective and its derivatives.

This paper contributes to this rapidly expanding field by presenting a stochastic algorithm based on negative curvature and Newton-type directions, computed for a subsampling model of the objective. A line-search technique is used to enforce suitable decrease for this model, and for a sufficiently large sample, a similar amount of reduction holds for the true objective. By using probabilistic reasoning, we can then obtain worst-case complexity guarantees for our framework, leading us to discuss appropriate notions of stationarity in a subsampling context. Our analysis, which we illustrate through real data experiments, encompasses the full sampled regime as a special case: it thus provides an insightful generalization of second-order line-search paradigms to subsampled settings.

Keywords: Nonconvex optimization, finite-sum problems, subsampling methods, negative curvature, worst-case complexity.

1 Introduction

In this paper, we aim to solve

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

*MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-en-Josas, France (elhoucine.bergou@inra.fr).

†ISAE-SUPAERO, Université de Toulouse, 31055 Toulouse Cedex 4, France (youssef.diouane@isae.fr).

‡Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague (vyacheslav.kungurtsev@fel.cvut.cz). Support for this author was provided by the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

§Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N Orchard St, Madison, WI 53715, USA (croyer2@wisc.edu). Support for this author was provided by Subcontract 3F-30222 from Argonne National Laboratory.

where the objective function f is not necessarily convex and the components f_i are assumed to be twice-continuously differentiable on $\mathbb{R}^n$. We are interested in the cases in which the number of components $N \geq 1$ is extremely large: for this reason, we will assume that it is computationally infeasible to evaluate the entire function, its gradient or its Hessian.

To overcome this issue, we consider the use of *subsampling techniques* to compute stochastic estimates of the objective function, its gradient and its Hessian. Given a random set $\mathcal{S}$ sampled from $\{1, \ldots, N\}$ and a point $x \in \mathbb{R}^n$, we will use

$$m(x; \mathcal{S}) := \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} f_i(x), \quad g(x; \mathcal{S}) := \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla f_i(x), \quad \text{and} \quad H(x; \mathcal{S}) := \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla^2 f_i(x), \quad (2)$$

to estimate the quantities $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$, respectively, where $|\mathcal{S}|$ denotes the cardinal of the sampling set $\mathcal{S}$. We are interested in iterative minimization processes that use a different, randomly selected sampling set $\mathcal{S}$ at every iteration.

The standard subsampling optimization procedure is the stochastic gradient descent (SGD) method, wherein the gradient is estimated by a subsampled (also called mini-batch) gradient and a step is taken in the negative of this direction. This is still the standard approach in many applications, including large-scale machine learning problems, particularly those arising from the training of deep neural net architectures [8]. However, SGD is known to be sensitive to nonconvexity, particularly in the context of training deep neural nets. It has indeed been observed that the optimization landscape for the associated (nonconvex) objective exhibits a significant number of saddle points, around which the flatness of the function tended to slow down the convergence of SGD [15]. Note that first-order methods almost never converge to saddle points [22], but are generally slowed down around the neighborhood of such points. One possible way to tackle this issue is to incorporate second-order information so as to guarantee that saddle points can be escaped from at a favorable rate. Various algorithms that provide such guarantees while only requiring gradient or Hessian-vector products have also been proposed, see for instance [1, 2, 23, 32]. Under certain accuracy conditions, which can be satisfied with arbitrarily high probability by controlling the size of the sample, these methods produce a sequence of iterates that converge to a local minimizer at a certain rate. Alternatively, one can extract second-order information and escape saddle points using accelerated gradient techniques in the stochastic setting [30]. The results are also in high probability, with a priori tuned step-sizes. Noise can be used to approximate second-order information as well [32]. Recent proposals [31, 33] derive high probability convergence results with second-order steps (e.g., Newton steps) based on sampled derivatives, by means of trust-region and cubic regularization frameworks.

Subsampling can be viewed as a particular case of stochastic optimization where the underlying distribution takes values within a discrete set. In the general stochastic optimization setting, a variety of algorithms have been extended to handle access to (sub)sampled derivatives, and possibly function values: of particular interest to us are the algorithms endowed with complexity guarantees. When the function values are assumed to be exactly available or are not used, one can employ strategies based on line search [11], cubic regularization [11, 19] or trust-region paradigms [14, 18] to compute a step of suitable length. Many algorithms building on SGD require the tuning of the step size parameter (also called learning rate), which can be cumbersome without knowledge of the Lipschitz constant. On the contrary, methods that are based on a globalization technique (line search, trust region, quadratic or cubic regularization) can control the size of the step in an adaptive way, and are thus less sensitive to parameter tuning.

In spite of their attractive properties with respect to the step size, globalized techniques are challenging to extend to the context of inexact function values. Indeed, these methods traditionally accept new iterates only if they produce a sufficient reduction of the objective value. Nevertheless, inexact variants of these schemes have been a recent topic of interest in the literature. In the context of stochastic optimization, several algorithms that explicitly deal with computing stochastic estimates of the function values, typically of trust-region type [6, 13, 21], have been proposed. In the specific case of least-squares problems, both approaches (exact and inexact function values) have been incorporated within a Levenberg-Marquardt framework [5, 4]. The use of stochastic function estimates in a line-search framework (a process that heavily relies on evaluating the function at tentative points) has also been the subject of very recent investigation. A study based on proprietary data [20] considered an inexact Newton and negative curvature procedure using each iteration's chosen mini-batch as the source of function evaluation sample in the line search. A more mature approach for line search was presented in [24], where extensive experiments matching performance to pre-tuned SGD were presented. An innovating technique based on a backtracking procedure for steps generated by a limited memory BFGS method for nonconvex problems using first order information was recently proposed in [7], and first-order convergence results were derived. Finally, during the final stages of preparation of this manuscript, a report presenting a stochastic line-search method was released [25]. Similarly to our scheme, this algorithm computes stochastic estimates for function and gradient values, which are then used within a line-search algorithm. However, the two methods differ in their inspiration and results: we provide more details about these differences in the next paragraph, and throughout the paper when relevant.

In this paper, we propose a line-search scheme with second-order guarantees based on subsampling function and derivatives. Our method uses these subsampled values to compute Newton-type and negative curvature steps. Although our framework bears similarities with the approach of [25], the two algorithms are equipped with different analyzes, each based on their own arguments from probability theory. The method of [25] is designed with first-order guarantees in mind (in particular, the use of negative curvature is not explored), and its complexity results are particularized to the nonconvex, convex and strongly convex cases; our work presents a line-search method that is dedicated to the nonconvex setting, and to the derivation of second-order results. As such, we are able to show a rate of convergence to points satisfying approximate second-order optimality conditions, in expectation. By contrast, earlier work on second-order guarantees for subsampling methods often focused on complexity bounds holding with a high probability (of accurate samples being taken at each iteration), disallowing poor outlier estimates of the problem function. We believe that our results form a complementary and interesting addition to the state-of-the-art literature on stochastic optimization methods.

We organize this paper as follows. In Section 2, we describe our proposed approach based on line-search techniques. In Section 3, we derive bounds on the amount of expected decrease that can be achieved at each iteration by our proposed approach. Section 4 gives the global convergence rate of our method under appropriate assumptions, followed by a discussion about the required properties and their satisfaction in practice. Numerical illustrations are provided in Section 5. A discussion of conclusions and future research is given in Section 6.

Throughout the paper, $\|.\|$ denotes the Euclidean norm. A vector $v \in \mathbb{R}^n$ will be called a unit vector if $\|v\| = 1$. Finally, $\mathbb{I}_n$ denotes the identity matrix of size n.

2 Subsampling line-search method

This section presents a line-search algorithm for solving the unconstrained optimization problem (1). This framework is based upon the exact version of the method proposed by Royer and Wright [29], but differs in several ways. Unlike [29], we do not consider the use of gradient-based steps. Although those could be added to our method without affecting the complexity guarantees, our purpose here is to advocate for the use of Newton and negative curvature directions; in addition, the upcoming analysis will be simplified with only three different forms of direction to consider (as opposed to five for the algorithm of [29]). As we will see in the next section, our analysis generalizes part of that of [29] to the case of subsampled functions and derivatives.

A detailed description of the proposed scheme is given in Algorithm 1. At each iteration k, our method computes a random sampling set $\mathcal{S}_k$, and the associated model $m_k(\cdot) := m(\cdot; \mathcal{S}_k)$. Based on $g_k := g(x_k; \mathcal{S}_k)$ and $H_k := H(x_k; \mathcal{S}_k)$, a search direction d_k is computed. The rule for choosing the form of the direction is slightly different from [29], in that it involves the gradient norm in addition to a tolerance on the minimum eigenvalue. As the results of Section 4 will show, in the fully sampled case, both rules lead to similar complexity guarantees.

Once the search direction has been determined, a backtracking line-search strategy is then applied to determine a suitable step size α_k that decreases the model by a sufficient amount (see condition (7)). This condition is instrumental in obtaining good complexity properties.

Two comments about the description of Algorithm 1 are in order. First, we observe that the method as stated is not equipped with a stopping criterion. Apart from budget considerations, one might be tempted to stop the method when the derivatives are suggesting that it is a second-order stationary point. However, since we only have access to subsampled versions of those derivatives, it is possible that we have not reached a stationary point for the true function. As we will establish later in the paper, one needs to take into account the accuracy of the model, and it might take several iterations to guarantee that we are indeed at a stationary point. We thus defer the proper discussion on stopping criteria and stationarity to Section 4.

Our second remark relates to the computation of a step. When the subsampled gradient g_k is zero and the subsampled matrix H_k is positive definite, we cannot compute a descent step using first or second-order information, as the current iterate is second-order stationary for the subsampled model. In that situation, and for the reasons mentioned above, we do not stop our method, but rather take a zero step and move to a new iteration and a new sample set. As we will show in Section 4, after a certain number of such iterations, one can guarantee that a stationary point has been reached, with high probability.

3 Expected decrease guarantees with subsampling

In this section, we derive bounds on the amount of expected decrease at each iteration. In doing so, we separate the cases of good and bad approximations. In the case of good approximations, we are able to guarantee decrease in the objective for every step taken in Algorithm 1. By controlling the sample size, one can adjust the probability of having a sufficiently good model, so that the guaranteed decrease for good approximations will compensate a possible increase for bad approximations on average.

Algorithm 1: A Line-search Algorithm based on Subsampling (ALAS).

Initialization: Choose $x_0 \in \mathbb{R}^n$, $\theta \in (0,1), \eta > 0$, $\epsilon > 0$.
for $k = 0, 1, ...$ **do**

1. Draw a random sample set $\mathcal{S}_k \subset \{1, \ldots, N\}$, and compute the associated quantities $g_k := g(x_k; \mathcal{S}_k), H_k := H(x_k; \mathcal{S}_k)$. Form the model:

$$m_k(x_k + s) := m(x_k + s; \mathcal{S}_k). \tag{3}$$

2. Compute λ_k as the minimum eigenvalue of the Hessian estimate H_k.
If $\lambda_k \geq -\epsilon^{1/2}$ and $\|g_k\| = 0$ set $\alpha_k = 0,\ d_k = 0$ and go to Step 7.

3. If $\lambda_k < -\epsilon^{1/2}$, compute a negative eigenvector v_k such that

$$H_k v_k = \lambda_k v_k,\ \|v_k\| = -\lambda_k,\ v_k^\top g_k \leq 0, \tag{4}$$

set $d_k = v_k$ and go to the line-search step.

4. If $\lambda_k > \|g_k\|^{1/2}$, compute a Newton direction d_k solution of

$$H_k d_k = -g_k, \tag{5}$$

go to the line-search step.

5. If d_k has not yet been chosen, compute it as a regularized Newton direction, solution of

$$\left(H_k + (\|g_k\|^{1/2} + \epsilon^{1/2})\mathbb{I}_n\right) d_k = -g_k, \tag{6}$$

and go to the line-search step.

6. **Line-search step** Compute the minimum index j_k such that the step length $\alpha_k := \theta^{j_k}$ satisfies the decrease condition:

$$m_k(x_k + \alpha_k d_k) - m_k(x_k) \ \leq\ -\frac{\eta}{6}\alpha_k^3 \|d_k\|^3. \tag{7}$$

7. Set $x_{k+1} = x_k + \alpha_k d_k$.

8. Set $k = k + 1$.

end

3.1 Preliminary assumptions and definitions

Throughout the paper, we will study Algorithm 1 under the following assumptions.

Assumption 3.1 *The function f is bounded below by $f_{\text{low}} \in \mathbb{R}$.*

Assumption 3.2 *The functions f_i are twice continuously differentiable with Lipschitz continuous gradients and Hessians, of respective Lipschitz constants L_i and $L_{H,i}$.*

A consequence of Assumption 3.2 is that f is twice continuously differentiable with Lipschitz continuous first and second-order derivatives. This property also holds for $m(\cdot; \mathcal{S})$, regardless of the value of $\mathcal{S}$ (we say that the property holds *for all realizations of* $\mathcal{S}$). In what follows, we will always consider that $m(\cdot; \mathcal{S})$ and f have L-Lipschitz continuous gradients and L_H-Lipschitz continuous Hessians, where

$$L := \max_i L_i, \qquad L_H := \max_i L_{H,i}.$$

Assumption 3.3 *There exists a compact set such that for any realization of the algorithm, the iterates are contained in this compact set.*

By Assumption 3.3, for the sequence of iterates $\{x_k\}$ generated by Algorithm 1, there exists a finite positive number U_g such that

$$U_g \geq \max_k \max_{i=1,\dots,N} \|\nabla f_i(x_k)\|.$$

The constant U_g can be used to bound the norms of the objective gradient and the model gradient at every iteration k, for any realization of the sample set sequence $\{\mathcal{S}_k\}$.

We define in a similar fashion a finite positive constant U_H such that:

$$U_H \geq \max_k \max_{i=1,\dots,N} \|\nabla^2 f_i(x_k)\|.$$

Finally, in what follows, for every iteration k, we let

$$\pi_k := \frac{|\mathcal{S}_k|}{N}$$

denote the *sample fraction* used at every iteration. Our objective is to determine conditions on this fraction that allow for decrease in expectation.

Whenever the sample sets in Algorithm 1 are drawn at random, the subsampling process introduces randomness in an iterative fashion at every iteration. As a result, Algorithm 1 results in a stochastic process $\{x_k, d_k, \alpha_k, g_k, H_k, m_k(x_k), m_k(x_k + \alpha_k d_k)\}$ (we point out that the sample fractions need not be random). To keep the notation of the paper simpler, we will use these notations for the random variables and their realizations. Most of our analysis will be concerned with random variables, but we will explicitly mention that realizations are considered when needed. Our goal is to show that under certain conditions on the sequences $\{g_k\}$, $\{H_k\}$, $\{m_k(x_k)\}$, $\{m_k(x_k + \alpha_k d_k)\}$ the resulting stochastic process has desirable convergence properties in expectation.

Inspired by a number of definitions in the model-based literature for stochastic or subsampled methods [3, 13, 21, 23], we introduce a notion of sufficient accuracy for our model function and its derivatives.

Definition 3.1 *Given a realization of Algorithm 1 and an iteration index k, the model $m_k : \mathbb{R}^n \mapsto \mathbb{R}$ is said to be* $(\delta_f, \delta_g, \delta_H)$-accurate *with respect to* (f, x_k, α_k, d_k) *when*

$$|f(x_k) - m_k(x_k)| \leq \delta_f \quad \textit{and} \quad |f(x_k + \alpha_k d_k) - m_k(x_k + \alpha_k d_k)| \leq \delta_f, \tag{8}$$

$$\|\nabla f(x_k) - g_k\| \leq \delta_g \quad \textit{and} \quad \|\nabla f(x_k + \alpha_k d_k) - g(x_k + \alpha_k d_k, \mathcal{S}_k)\| \leq \delta_g, \tag{9}$$

as well as

$$\|\nabla^2 f(x_k) - H_k\| \leq \delta_H, \tag{10}$$

where $g_k := \nabla m_k(x_k)$, $H_k := \nabla^2 m_k(x_k)$ and $\delta_f, \delta_g, \delta_H$ are nonnegative constants.

Condition (8) is instrumental in establishing decrease guarantees for our method, while conditions (9) and (10) play a key role in defining proper notions of stationarity (see Section 4). Since we are operating with a sequence of random samples and models, we need a probabilistic equivalent of Definition 3.1, which is given below.

Definition 3.2 *Let $p \in (0, 1]$, $\delta_f \geq 0$, $\delta_g \geq 0$ and $\delta_H \geq 0$. A sequence of functions $\{m_k\}_k$ is called p-probabilistically $(\delta_f, \delta_g, \delta_H)$-accurate for Algorithm 1 if the events*

$$I_k := \{m_k \textit{ is } (\delta_f, \delta_g, \delta_H)\textit{-accurate with respect to } (f, x_k, \alpha_k, d_k)\}$$

satisfy

$$p_k := \mathbb{P}\left(I_k | \mathcal{F}_{k-1}\right) \geq p,$$

where $\mathcal{F}_{k-1}$ is the σ-algebra generated by the sample sets $\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{k-1}$, and we define $\mathbb{P}(I_0|\mathcal{F}_{-1}) := \mathbb{P}(I_0)$.

Observe that if we sample the full data at every iteration (that is, $\mathcal{S}_k = \{1, \ldots, N\}$ for all k), the resulting model sequence satisfies the above definition for any $p \in [0, 1]$ and any positive values $\delta_f, \delta_g, \delta_H$. Given our choice of model (2), the accuracy properties are directly related to the random sampling sets; however, to stay consistent with the existing literature, we will talk about accuracy of the models. In particular, we will express conditions for good convergence behavior based on the sample size rather than the probability of accuracy.

In the rest of the paper, we assume that the estimate functions of the problem form a probabilistically accurate sequence as follows.

Assumption 3.4 *The sequence $\{m_k\}_k$ produced by Algorithm 1 is p-probabilistically δ-accurate, with $\delta := (\delta_f, \delta_g, \delta_H)$ and $p \in (0, 1]$.*

We now introduce the two notions of stationarity that will be considered in our analysis.

Definition 3.3 *Consider a realization of Algorithm 1, and let ϵ_g, ϵ_H be two positive tolerances. We say that the k-th iterate x_k is* (ϵ_g, ϵ_H)-model stationary *if*

$$\min\left\{\|g_k\|, \|g(x_{k+1}, \mathcal{S}_k)\|\right\} \leq \epsilon_g \quad \textit{and} \quad \lambda_k \geq -\epsilon_H. \tag{11}$$

Similarly, we will say that x_k is (ϵ_g, ϵ_H)-function stationary *if*

$$\min\left\{\|\nabla f(x_k)\|, \|\nabla f(x_{k+1})\|\right\} \leq \epsilon_g \quad \text{or} \quad \lambda_{\min}(\nabla^2 f(x_k)) \geq -\epsilon_H. \tag{12}$$

Note that the two definitions above are equivalent whenever the model consists of the full function. We also observe that the definition of model stationarity involves the norm of the vector

$$g_k^+ := g(x_k + \alpha_k d_k; \mathcal{S}_k). \tag{13}$$

The norm of this "next gradient" is a major tool for the derivation of complexity results in Newton-type methods [10, 29]. In a subsampled setting, a distinction between g_k^+ and g_{k+1} has to be made because these two vectors are computed using different sample sets.

Our objective is to guarantee convergence towards a point satisfying a function stationarity property (12), yet we will only have control on achieving model stationarity. The accuracy of the models will be instrumental in relating the two properties, as shown by the lemma below.

Lemma 3.1 *Let Assumptions 3.1 and 3.2 hold. Consider a realization of the method that reaches an iterate x_k such that x_k is (ϵ_g, ϵ_H)-model stationary. Suppose further that the model m_k is $(\delta_f, \delta_g, \delta_H)$-accurate with*

$$\delta_g \leq \kappa_g \epsilon_g \quad \text{and} \quad \delta_H \leq \kappa_H \epsilon_H \tag{14}$$

where κ_g and κ_H are positive, deterministic constants independent of k. Then, x_k is a $((1+\kappa_g)\epsilon_g, (1+\kappa_H)\epsilon_H)$-function stationary point.

Proof. Let x_k be an iterate such that

$$\min\{\|g_k\|, \|g_k^+\|\} \leq \epsilon_g \quad \text{and} \quad \lambda_k \geq -\epsilon_H.$$

Looking at the first property, suppose that $\|g_k\| \leq \epsilon_g$. In that case, we have:

$$\|\nabla f(x_k)\| \leq \|\nabla f(x_k) - g_k\| + \|g_k\| \leq \delta_g + \epsilon_g \leq (\kappa_g + 1)\epsilon_g.$$

A similar reasoning shows that if $\|g_k^+\| \leq \epsilon_g$, we obtain $\|\nabla f(x_{k+1})\| \leq (\kappa_g + 1)\epsilon_g$; thus, we must have

$$\min\left\{\|\nabla f(x_k)\|, \|\nabla f(x_{k+1})\|\right\} \leq (1+\kappa_g)\epsilon_g.$$

Consider now a unit eigenvector v for $\nabla^2 f(x_k)$ associated with $\lambda_{\min}(\nabla^2 f(x_k))$, one has

$$\begin{aligned}
\lambda_k - \lambda_{\min}(\nabla^2 f(x_k)) &\leq v^\top H_k v - v^\top \nabla^2 f(x_k) v \\
&\leq \|H_k - \nabla^2 f(x_k)\| \|v\|^2 \\
&\leq \delta_H.
\end{aligned}$$

Hence, by using (14), one gets

$$\lambda_{\min}(\nabla^2 f(x_k)) = \big(\lambda_{\min}(\nabla^2 f(x_k)) - \lambda_k\big) + \lambda_k \geq -\delta_H - \epsilon_H \geq -(\kappa_H + 1)\epsilon_H.$$

Overall, we have shown that

$$\min\left\{\|\nabla f(x_k)\|, \|\nabla f(x_{k+1})\|\right\} \leq (1+\kappa_g)\epsilon_g \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x_k)) \geq -(1+\kappa_H)\epsilon_H,$$

and thus x_k is also a $((1+\kappa_g)\epsilon_g, (1+\kappa_H)\epsilon_H)$-function stationary point. ∎

The reciprocal result of Lemma 3.1 will also be of interest to us in the next section.

Lemma 3.2 *Consider a realization of Algorithm 1 and the associated k-th iteration. Suppose that x_k is* not *$((1+\kappa_g)\epsilon_g, (1+\kappa_H)\epsilon_H)$-function stationary, and that the model m_k is $\delta = (\delta_f, \delta_g, \delta_H)$-accurate with $\delta_g \leq \kappa_g \epsilon_g$ and $\delta_H \leq \kappa_H \epsilon_H$ where κ_g and κ_H are positive constants. Then, x_k is* not *(ϵ_g, ϵ_H)-model stationary.*

3.2 A general expected decrease result

In this section, we study the guarantees that can be obtained (in expectation) for the various types of direction considered by our method. By doing so, we identify the necessary requirements on our sampling procedure, as well as on our accuracy threshold for the model values.

In what follows, we will make use of the following constants:

$$c_{nc} := \frac{3\theta}{L_H+\eta},\ c_n := \min\left\{\left[\frac{2}{L_H}\right]^{1/2}, \left[\frac{3\theta}{L_H+\eta}\right]\right\},\ c_{rn} := \min\left\{\frac{1}{1+\sqrt{1+L_H/2}}, \left[\frac{6\theta}{L_H+\eta}\right]\right\},$$

$$\bar{j}_{nc} := \left[\log_\theta\left(\frac{3}{L_H+\eta}\right)\right]_+,\ \bar{j}_n := \left[\log_\theta\left(\sqrt{\frac{3}{L_H+\eta}}\frac{\epsilon^{1/2}}{\sqrt{U_g}}\right)\right]_+,\ \bar{j}_{rn} := \left[\log_\theta\left(\frac{6}{L_H+\eta}\frac{\epsilon}{U_g}\right)\right]_+.$$

Those constants are related to the line-search steps that can be performed at every iteration of Algorithm 1. When the current iterate is not an approximate stationary point of the model, we can bound this number independently of k. This is the purpose of the following lemma.

Consider the event

$$E_k := \left\{ \min\{\|g_k\|, \|g_k^+\|\} > \epsilon \quad \text{or} \quad \lambda_k < -\epsilon^{1/2} \right\}, \tag{15}$$

where ϵ is the tolerance used in Algorithm 1.

Lemma 3.3 *Let Assumptions 3.1, 3.2 and 3.3 hold for a realization of Algorithm 1. Consider an iteration k such that the event E_k holds. Then, the backtracking line search terminates with the step length $\alpha_k = \theta^{j_k}$, with $j_k \leq \bar{j} + 1$ and*

$$\alpha_k \|d_k\| \ \geq\ c\,\epsilon^{1/2}, \tag{16}$$

where

$$c := \min\{c_{nc}, c_n, c_{rn}\} \tag{17}$$

and

$$\bar{j} := \max\{\bar{j}_{nc}, \bar{j}_n, \bar{j}_{rn}\}. \tag{18}$$

Proof. We consider in turn the three possible steps that can be taken at iteration k, and obtain a lower bound on the amount $\alpha_k\|d_k\|$ for each of those.

Case 1: $\lambda_k < -\epsilon^{1/2}$ (negative curvature step). In that case, we apply the same reasoning than in [29, Proof of Lemma 1] with the model m_k playing the role of the objective, the backtracking line search terminates with the step length $\alpha_k = \theta^{j_k}$, with $j_k \leq \bar{j}_{nc} + 1$ and $\alpha_k \geq \frac{3\theta}{L_H+\eta}$. When d_k is computed as a negative curvature direction, one has $\|d_k\| = -\lambda_k > 0$. Hence,

$$\alpha_k\|d_k\| \ \geq\ \frac{3\theta}{L_H+\eta}[-\lambda_k] \ =\ c_{nc}[-\lambda_k] \geq c\epsilon^{1/2}.$$

Case 2: $\lambda_k > \|g_k\|^{1/2}$ (Newton step). Because (15) holds and $\lambda_k > 0$ in this case, we necessarily have $\|\tilde{g}_k\| = \min\{\|g_k\|, \|g_k^+\|\} > \epsilon$. From Algorithm 1, we know that d_k is chosen as the Newton step. Hence, using the argument of [29, Proof of Lemma 3] with $\epsilon_H = \epsilon^{1/2}$, the backtracking line search terminates with the step length $\alpha_k = \theta^{j_k}$, where

$$j_k \leq \left[\log_\theta\left(\sqrt{\frac{3}{L+\eta}}\frac{\epsilon^{1/2}}{\sqrt{U_g}}\right)\right]_+ + 1 = \bar{j}_n + 1,$$

thus the first part of the result holds. If the unit step size is chosen, we have by [29, Relation (23)] that

$$\alpha_k\|d_k\| = \|d_k\| \geq \left[\frac{2}{L_H}\right]^{1/2} \|g(x_k + \alpha_k d_k; \mathcal{S}_k)\|^{1/2} \geq c\epsilon^{1/2}. \tag{19}$$

Consider now the case $\alpha_k < 1$. Using [29, Relations (25) and (26), p. 1457], we have:

$$\|d_k\| \geq \frac{3}{L_H + \eta}\epsilon_H = \frac{3}{L_H + \eta}\epsilon^{1/2}$$

and

$$\alpha_k \geq \theta\sqrt{\frac{3}{L_H + \eta}}\epsilon_H^{1/2}\|d_k\|^{-1/2} = \theta\sqrt{\frac{3}{L_H + \eta}}\epsilon^{1/4}\|d_k\|^{-1/2}.$$

As a result,

$$\begin{aligned}
\alpha_k\|d_k\| &\geq \theta\left[\frac{3}{L_H + \eta}\right]^{1/2}\epsilon^{1/4}\|d_k\|^{-1/2}\|d_k\| \\
\alpha_k\|d_k\| &\geq \left[\frac{3\theta}{L_H + \eta}\right]\epsilon^{1/2} \geq c\epsilon^{1/2}.
\end{aligned} \tag{20}$$

Case 3: (Regularized Newton step) This case occurs when the conditions for the other two cases fail, that is, when $-\epsilon^{1/2} \leq \lambda_k \leq \|g_k\|^{1/2}$. We again exploit the fact that (15) holds to deduce that we necessarily have $\|\tilde{g}_k\| = \min\{\|g_k\|, \|g_k^+\|\} > \epsilon$. This in turn implies that $\min\{\|\tilde{g}_k\|\epsilon^{-1/2}, \epsilon^{1/2}\} \geq \epsilon^{1/2}$. As in the proof of the previous lemma, we apply the theory of [29, Proof of Lemma 4] using $\epsilon_H = \epsilon^{1/2}$. We then know that the backtracking line search terminates with the step length $\alpha_k = \theta^{j_k}$, with

$$j_k \leq \left[\log_\theta\left(\frac{6}{L_H + \eta}\frac{\epsilon}{U_g}\right)\right]_+ + 1 = \bar{j}_{rn} + 1,$$

since $\epsilon_H = \epsilon^{1/2}$.

We now distinguish between the cases $\alpha_k = 1$ and $\alpha_k < 1$. If the unit step size is chosen, we can use [29, relations 30 and 31], where $\nabla f(x_k + d_k)$ and ϵ_H are replaced by g_k^+ and $\epsilon^{1/2}$, respectively. This gives

$$\alpha_k\|d_k\| = \|d_k\| \geq \frac{1}{1 + \sqrt{1 + L_H/2}}\min\left\{\|g_k^+\|/\epsilon^{1/2}, \epsilon^{1/2}\right\}.$$

Therefore, if the unit step is accepted, one has by [29, equation 31]

$$\begin{aligned}
\alpha_k\|d_k\| &\geq \frac{1}{1 + \sqrt{1 + L_H/2}}\min\left\{\|g_k^+\|\epsilon^{-1/2}, \epsilon^{1/2}\right\} \\
&\geq \frac{1}{1 + \sqrt{1 + L_H/2}}\min\left\{\|\tilde{g}_k\|\epsilon^{-1/2}, \epsilon^{1/2}\right\}.
\end{aligned} \tag{21}$$

Considering now the case $\alpha_k < 1$ and using [29, equation 32, p. 1459], we have:

$$\alpha_k \geq \theta\frac{6}{L_H + \eta}\epsilon_H\|d_k\|^{-1} = \frac{6\theta}{L_H + \eta}\epsilon^{1/2}\|d_k\|^{-1},$$

which leads to

$$\alpha_k\|d_k\| \;\geq\; \frac{6\theta}{L_H+\eta}\epsilon^{1/2}. \tag{22}$$

Putting (21) and (22) together, we obtain

$$\begin{aligned}\alpha_k\|d_k\| &\geq \min\left\{\frac{1}{(1+\sqrt{1+L_H/2})^3}, \left[\frac{6\theta}{L_H+\eta}\right]^3\right\}\min\{\|\tilde{g}_k\|\epsilon^{-1/2}, \epsilon^{1/2}\}\\ &= c_{rn}\min\{\|\tilde{g}_k\|\epsilon^{-1/2}, \epsilon^{1/2}\} \geq c\epsilon^{1/2}.\end{aligned}$$

By putting the three cases together, we arrive at the desired conclusion. ■

In order to tackle the worst-case behavior of our method, we provide the following upper bound on the norm of the search directions computed by our method.

Lemma 3.4 *Let Assumptions 3.2 and 3.3 hold for a realization of Algorithm 1. Then, for any index k, it holds that*

$$\|d_k\| \leq \max\{U_H, U_g^{1/2}\}. \tag{23}$$

Proof. The bound (23) trivially holds if $\|d_k\| = 0$, thus we only need to prove that it holds for $\|d_k\| > 0$. We consider three disjoint cases:

Case 1: $\lambda_k < -\epsilon^{1/2}$. Then the negative curvature step is taken and $\|d_k\| = |\lambda_k| \leq U_H$.

Case 2: $\lambda_k > \|g_k\|^{1/2}$. We can suppose that $\|g_k\| > 0$ because otherwise $\|d_k\| = 0$. Then, d_k is a Newton step with

$$\|d_k\| \leq \|H_k^{-1}\|\|g_k\| \leq \|g_k\|^{-1/2}\|g_k\| \leq \|g_k\|^{1/2} \leq U_g^{1/2}.$$

Case 3: $-\epsilon^{1/2} \leq \lambda_k \leq \|g_k\|^{1/2}$. As in Case 2, we suppose that $\|g_k\| > 0$ as $\|d_k\| = 0$ if this does not hold. Then, d_k is a regularized Newton step with

$$\begin{aligned}\|d_k\| &= \|(H_k + (\|g_k\|^{1/2}+\epsilon^{1/2})\mathbb{I}_n)^{-1}g_k\|\\ &\leq \|(H_k + (\|g_k\|^{1/2}+\epsilon^{1/2})\mathbb{I}_n)^{-1}\|\|g_k\|\\ &\leq (\lambda_k + \|g_k\|^{1/2}+\epsilon^{1/2})^{-1}\|g_k\|/\\ &\leq \|g_k\|^{-1/2}\|g_k\| \leq U_g^{1/2},\end{aligned}$$

where the last line uses the fact that $\lambda_k + \epsilon^{1/2} \geq 0$ and $\|g_k\| > 0$. ■

For future use in this section, we define the following function on $[0,\infty)\times[0,1]$:

$$\varrho(t,q) \;:=\; \frac{(1-q)U_L}{(1-q)U_L + q\frac{\eta t^3}{24}}, \quad \text{where } U_L := U_g\max\{U_H, U_g^{1/2}\} + \tfrac{L}{2}\max\{U_h^2, U_g\}. \tag{24}$$

We observe that the function ϱ is well-defined, with values in $[0,1]$, and decreasing in its first and second arguments.

Based on the result of Lemmas 3.3 and 3.4, we can provide a generic guarantee on the expected decrease at every iteration: this is the purpose of Theorem 3.1.

Theorem 3.1 *Let Assumptions 3.1 and 3.2 hold. Suppose also that Assumption 3.4 holds with* $\delta = (\delta_f, \delta_g, \delta_H)$ *satisfying*

$$\delta_f \leq \frac{\eta}{24} c^3 \epsilon^{3/2}, \quad \delta_g \leq \kappa_g \epsilon, \quad \delta_H \leq \kappa_H \epsilon^{1/2} \tag{25}$$

where $\epsilon > 0$, $\kappa_g \in (0,1)$, $\kappa_H \in (0,1)$ *and* c *is chosen as in Lemma 3.3. Finally, consider the following random event*

$$E_k^{\text{sta}} = \left\{ x_k \text{ is not } ((1+\kappa_g)\epsilon, (1+\kappa_H)\epsilon^{1/2})\text{-function stationary} \right\}. \tag{26}$$

If the sample fraction π_k *is chosen such that*

$$\pi_k \ \geq \ \varrho(c\epsilon^{1/2}, p), \tag{27}$$

where ϱ *is given by* (24)*, then*

$$\mathbb{E}\left[f(x_k + \alpha_k d_k) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \ \leq \ -\varrho(c\epsilon^{1/2}, p) \frac{\eta}{24} c^3 \epsilon^{3/2}. \tag{28}$$

Proof. By definition, one has that:

$$\begin{aligned}
&\mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \\
&= \mathbb{P}(I_k | \mathcal{F}_{k-1}, E_k^{\text{sta}}) \mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}, I_k \right] + \\
&\left(1 - \mathbb{P}(I_k | \mathcal{F}_{k-1}, E_k^{\text{sta}})\right) \mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}, \bar{I}_k \right] \\
&= \tilde{p}_k \mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}, I_k \right] + (1 - \tilde{p}_k) \mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}, \bar{I}_k \right]
\end{aligned} \tag{29}$$

in which I_k is the event corresponding to the model being δ-accurate and $\tilde{p}_k := \mathbb{P}(I_k | \mathcal{F}_{k-1}, E_k^{\text{sta}})$. We note that $\tilde{p}_k \geq p$, in fact since $\mathcal{F}_{k-1} \cap E_k^{\text{sta}} \subset \mathcal{F}_{k-1}$, we can apply the result from [16, Theorem 5.1.6], which states that for any random variable X (not necessarily belonging to one of the σ-algebras), we have

$$\mathbb{E}\left[X | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] = \mathbb{E}\left[\mathbb{E}\left[X | \mathcal{F}_{k-1} \right] | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \tag{30}$$

Thus, denoting by $\mathbf{1}(I_k)$ the indicator variable of the random event I_k, we have

$$\begin{aligned}
\tilde{p}_k = \mathbb{P}\left(I_k | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right) &= \mathbb{E}\left[\mathbf{1}(I_k) | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\mathbf{1}(I_k) | \mathcal{F}_{k-1} \right] | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \\
&= \mathbb{E}\left[\mathbb{P}(I_k | \mathcal{F}_{k-1}) | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] \\
&= \mathbb{E}\left[p_k | \mathcal{F}_{k-1}, E_k^{\text{sta}} \right] = p_k \geq p
\end{aligned}$$

We now first bound the term corresponding to the occurrence of I_k in (29). Under occurrence of I_k and E_k^{sta}, one has:

$$\begin{aligned}
&f(x_k + \alpha_k d_k) - f(x_k) \\
&= f(x_k + \alpha_k d_k) - m_k(x_k + \alpha_k d_k) + m_k(x_k + \alpha_k d_k) - m_k(x_k) + m_k(x_k) - f(x_k) \\
&\leq 2\delta_f + m_k(x_k + \alpha_k d_k) - m_k(x_k) \\
&\leq 2\delta_f - \frac{\eta}{6} \alpha_k^3 \|d_k\|^3 \\
&\leq \frac{\eta c^3}{12} \epsilon^{3/2} - \frac{\eta}{6} \alpha_k^3 \|d_k\|^3 \ \leq \ -\frac{\eta}{12} \alpha_k^3 \|d_k\|^3 \leq -\frac{\eta c^3}{12} \epsilon^{3/2} \ \leq \ -\varrho(c\epsilon^{1/2}, p) \frac{\eta c^3}{12} \epsilon^{3/2},
\end{aligned} \tag{31}$$

where we used the bound (25) and the fact that $\alpha_k\|d_k\| \geq c\,\epsilon^{1/2}$, which follows from Lemma 3.3 and E_k^{sta}.

We now turn to the second case in (29), for which we exploit the following decomposition:

$$f(x_k+\alpha_k d_k) - f(x_k) = \pi_k \left(m_k(x_k+\alpha_k d_k) - m_k(x_k)\right) + (1-\pi_k)\left(f_{\mathcal{S}_k^c}(x_k+\alpha_k d_k) - f_{\mathcal{S}_k^c}(x_k)\right),$$

with $f_{\mathcal{S}_k^c} = \frac{1}{N-|\mathcal{S}_k|}\sum_{i\notin\mathcal{S}_k} f_i$. Using the decrease condition (7) to bound the first term, and a first-order Taylor expansion to bound the second term, we obtain:

$$\begin{aligned}
&f(x_k+\alpha_k d_k) - f(x_k) \\
&\leq -\pi_k\frac{\eta}{6}\alpha_k^3\|d_k\|^3 + (1-\pi_k)\,\frac{1}{N-|\mathcal{S}_k|}\sum_{i\notin\mathcal{S}_k}\left\{\alpha_k\nabla f_i(x_k)^\top d_k + \frac{L_i}{2}\alpha_k^2\|d_k\|^2\right\} \\
&\leq -\pi_k\frac{\eta}{6}\alpha_k^3\|d_k\|^3 + (1-\pi_k)\max_{i\notin\mathcal{S}_k}\left\{\alpha_k\nabla f_i(x_k)^\top d_k + \frac{L_i}{2}\alpha_k^2\|d_k\|^2\right\} \\
&\leq -\pi_k\frac{\eta}{6}\alpha_k^3\|d_k\|^3 + (1-\pi_k)\left\{\alpha_k\|\nabla f_{i_k}(x_k)\|\|d_k\| + \frac{L_{i_k}}{2}\alpha_k^2\|d_k\|^2\right\},
\end{aligned}$$

where $i_k \in \arg\max_{i\notin\mathcal{S}_k}\left\{\alpha_k\nabla f_i(x_k)^\top d_k + \frac{L_i}{2}\alpha_k^2\|d_k\|^2\right\}$ and we used the Lipschitz continuity assumption on the functions f_i's. By Assumption 3.3, we further have:

$$f(x_k+\alpha_k d_k) - f(x_k) \;\leq\; -\pi_k\frac{\eta}{6}\alpha_k^3\|d_k\|^3 + (1-\pi_k)\left\{U_g\alpha_k\|d_k\| + \frac{L}{2}\alpha_k^2\|d_k\|^2\right\}.$$

Using now Lemma 3.4, we get

$$\begin{aligned}
f(x_k+\alpha_k d_k) - f(x_k) &\leq (1-\pi_k)\left\{U_g\max\{U_H, U_g^{1/2}\} + \frac{L}{2}\max\{U_h^2, U_g\}\right\} \\
&= (1-\pi_k)\,U_L \leq \Big(1-\varrho(c\epsilon^{1/2},p)\Big)\,U_L,
\end{aligned}$$

where U_L is defined as in (24), and we use (27) to obtain the last inequality.
Putting the two cases together then yields:

$$\mathbb{E}\left[f(x_k+\alpha_k d_k) - f(x_k)|\mathcal{F}_{k-1}, E_k^{\text{sta}}\right] \;\leq\; -\tilde{p}_k\,\varrho(c\epsilon^{1/2},p)\frac{\eta c^3}{12}\epsilon^{3/2} + (1-\tilde{p}_k)\Big(1-\varrho(c\epsilon^{1/2},p)\Big)\,U_L.$$

Hence,

$$\mathbb{E}\left[f(x_k+\alpha_k d_k) - f(x_k)\mid \mathcal{F}_{k-1}, E_k^{\text{sta}}\right] \;\leq -\,\varrho(c\epsilon^{1/2},p)\tfrac{\eta c^3}{24}\epsilon^{3/2}. \tag{32}$$

provided that

$$\varrho(c\epsilon^{1/2},p) = \frac{(1-p)U_L}{(1-p)U_L + p\frac{\eta c^3}{24}\epsilon^{3/2}} \;\geq\; \frac{(1-\tilde{p}_k)U_L}{(1-\tilde{p}_k)U_L + \tilde{p}_k\frac{\eta c^3}{24}\epsilon^{3/2}} = \varrho(c\epsilon^{1/2},\tilde{p}_k),$$

and this is true because the expression on the right-hand side is a decreasing function in its second argument. The desired result follows. ■

Note that conditions (25) and (27) are always satisfied when the whole data is sampled, regardless of the value of p or ϵ ($\pi_k = 1$ and one can choose $\delta = (0,0,0)$). Note also that by

setting $p = \pi_k = 1$, we recover the deterministic result derived in [29] up to a constant factor[1] and the lower bound (27) becomes superfluous.

In the next section, we will see how the result of Theorem 3.1 can be used to derive a global convergence rate for Algorithm 1.

4 Global convergence rate and complexity analysis

In this section, we provide a global rate of convergence towards an approximate stationary point in expectation for our method. More precisely, we are seeking an $((1+\kappa_g)\epsilon, (1+\kappa_H)\epsilon^{1/2})$-stationary point in the sense of Definition 3.3, that is, an iterate satisfying:

$$\min\{\|\nabla f(x_k)\|, \|\nabla f(x_{k+1})\|\} \leq (1+\kappa_g)\epsilon \quad \text{and} \quad \lambda_{\min}(\nabla^2 f(x_k)) \geq -(1+\kappa_H)\epsilon^{1/2}. \tag{33}$$

Since our method only operates with a (subsampling) model of the objective, we are only able to observe if the current iterate is an $(\epsilon, \epsilon^{1/2})$-model stationary point according to Definition 3.3, that is, an iterate x_k such that

$$\min\{\|g_k\|, \|g_k^+\|\} \leq \epsilon \quad \text{and} \quad \lambda_k \geq -\epsilon^{1/2}.$$

Compared to the general setting of Definition 3.3, we are using $\epsilon_g = \epsilon_H^2 = \epsilon$. This specific choice of first- and second-order tolerances has been observed to yield optimal complexity bounds for a number of algorithms, in the sense that the dependence on ϵ is minimal (see e.g. [9, 29]). The decision between the various choices of directions in Algorithm 1 was also made with these choices in mind.

Our goal is to relate the model stationarity property (4) to its function stationarity counterpart (33). For this purpose, we first establish a general result regarding the expected number of iterations required to reach function stationarity. We then present a stopping criterion involving multiple consecutive iterations of model stationarity: using such a criterion allows to stop the algorithm at a function stationary point with high probability. Moreover, we show that the expected number of iterations until this stopping occurs is of the same order of magnitude than the expected number of iterations required to reach a function stationary point.

4.1 Expected iteration complexity

In addition to the results from the previous section, the proof of our expected complexity bound will rely on two arguments from martingales and stopping time theory. The first one is a martingale convergence result [26, Theorem 1], adapted to our setting in Theorem 4.1.

Theorem 4.1 *Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space. Let $\{\Sigma^k\}_k$ be a sequence of sub-sigma algebras of Σ such that $\Sigma^k \subset \Sigma^{k+1}$.*

If α^k is a positively valued sequence of random variables on Σ, and there is a deterministic sequence $\nu^k \geq 0$ such that,

$$\mathbb{E}[\alpha^{k+1}|\Sigma^k] + \nu^k \leq \alpha^k$$

then α^k converges to a $[0,\infty)$-valued random variable almost surely and $\sum_k \nu^k < \infty$.

[1] This constant factor of $\frac{1}{4}$ arises from both the overestimation of δ_f by a positive constant and the compensation of the "bad" case in which the model is inaccurate.

At each iteration, Theorem 3.1 guarantees a certain expected decrease for the objective function. Theorem 4.1 will be used to show that such a decrease cannot hold indefinitely if the objective is bounded from below.

The second argument comes from stopping time analysis (see, e.g., [28, Theorem 6.4.1]) and is given in Theorem 4.2. The notations have been adapted to our setting.

Theorem 4.2 *Let T be a stopping time for the process $\{Z_k, k \geq 0\}$ and let $\bar{Z}_k = Z_k$ for $k \leq T$ and $\bar{Z}_k = Z_T$ for $k > T$. If either one of the three properties below hold:*

- *$\bar{Z}_k$ is uniformly bounded;*
- *T is bounded;*
- *$\mathbb{E}[T] < \infty$ and there is an $R < \infty$ such that $\mathbb{E}[|Z_{k+1} - Z_k| \,|\, Z_0, ..., Z_k] < R$;*

Then, $\mathbb{E}\left[\bar{Z}_k\right] \to \mathbb{E}\left[Z_T\right]$. Moreover, $\mathbb{E}[Z_T] \geq \mathbb{E}[Z_0]$ (resp. $\mathbb{E}[Z_T] \leq \mathbb{E}[Z_0]$) if $\{Z_k\}$ is a submartingale (resp. a supermartingale).

Theorem 4.2 enables us to exploit the martingale-like property of Definition 3.2 in order to characterize the index of the first stationary point encountered by the method.

Using both theorems along with Theorem 3.1, we bound the expected number of iterations needed by Algorithm 1 to produce an approximate stationary point for the model.

Theorem 4.3 *Let Assumptions 3.1, 3.2, 3.3 and 3.4 hold, with $\delta = (\delta_f, \delta_g, \delta_H)$ satisfying* (25)*. Suppose that for every index k, the sample size π_k satisfies* (27)*.*

Define T_ϵ as the first iteration index k of Algorithm 1 for which

$$\min\{\|\nabla f(x_k)\|, \|\nabla f(x_{k+1})\|\} \leq (1+\kappa_g)\epsilon \quad \text{and} \quad \lambda_{\min}\left(\nabla^2 f(x_k)\right) \geq -(1+\kappa_H)\epsilon^{1/2}.$$

Then, $T_\epsilon < \infty$ almost surely, and

$$\mathbb{E}[T_\epsilon] \;\leq\; \begin{cases} \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\epsilon^{-3/2} + 1 & \text{if } \pi_k = 1 \;\forall k; \\ \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\, \varrho(c\epsilon^{1/2}, p)^{-1}\, \epsilon^{-3/2} + 1 & \text{otherwise,} \end{cases} \tag{34}$$

where $\hat{c} = \frac{\eta}{24}c^3$.

Proof. Since both x_{k+1} and $\mathcal{S}_k$ belong to $\mathcal{F}_k$, we have $\{T_\epsilon = k\} \in \mathcal{F}_k$ for all k, thus T_ϵ is indeed a stopping time.

We first show that $\{T_\epsilon = \infty\}$ has a zero probability of occurrence. To this end, suppose that for every iteration index k, we have $k < T_\epsilon$. In particular, this means that the events $E_0^{\text{sta}}, \ldots, E_k^{\text{sta}}$ occur, where we recall that $E_j^{\text{sta}} \in \mathcal{F}_{j-1}$. We thus define the following filtration:

$$\mathcal{T}_0 = \mathcal{F}_{-1} \cap E_0^{\text{sta}}, \quad \mathcal{T}_k = \mathcal{F}_{k-1} \cap \left(E_0^{\text{sta}} \cap \cdots \cap E_k^{\text{sta}}\right) \;\forall k \geq 1, \tag{35}$$

where we use $\mathcal{F} \cap E$ to denote the trace σ-algebra of the event E on the σ-algebra $\mathcal{F}$, i.e., $\mathcal{F} \cap E = \{E \cap F \;:\; F \in \mathcal{F}\}$. For every $k \geq 0$ and any event A, we thus have by the same argument used to establish (30) that

$$\mathbb{E}\left[A|\mathcal{T}_k\right] = \mathbb{E}\left[\mathbb{E}\left[A|\mathcal{F}_{k-1}, E_k^{\text{sta}}\right]|\mathcal{T}_k\right]$$

and we also have $\mathcal{T}_k \subset \mathcal{T}_{k+1}$.

For notational convenience, we define

$$c_\epsilon = \begin{cases} \hat{c}\epsilon^{3/2} & \text{if } \pi_k = 1 \ \forall k; \\ \hat{c}\, \varrho(c\epsilon^{1/2}, p)\epsilon^{3/2} & \text{otherwise.} \end{cases} \tag{36}$$

If $T_\epsilon = \infty$, then the assumptions of Theorem 3.1 are satisfied for all iterations k. In particular, for all k, event E_k^{sta} occurs and $\mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}\right] \leq -c_\epsilon$. Recalling the definition of $\mathcal{T}_k$ (35), we get,

$$\mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{T}_k\right] \;=\; \mathbb{E}\left[\mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{F}_{k-1}, E_k^{\text{sta}}\right] \mid \mathcal{T}_k\right] \;\leq\; \mathbb{E}\left[-c_\epsilon \mid \mathcal{T}_k\right] = -c_\epsilon.$$

Thus,

$$\begin{aligned} \mathbb{E}\left[f(x_{k+1}) - f(x_k) \mid \mathcal{T}_k\right] &\leq -c_\epsilon \\ \mathbb{E}\left[f(x_{k+1}) \mid \mathcal{T}_k\right] + c_\epsilon &\leq \mathbb{E}\left[f(x_k) \mid \mathcal{T}_k\right] \\ \mathbb{E}\left[f(x_{k+1}) \mid \mathcal{T}_k\right] + c_\epsilon &\leq f(x_k), \end{aligned}$$

where the last relation comes from the fact that $x_k \in \mathcal{T}_k$. Subtracting f_{low} on both sides yields:

$$\mathbb{E}\left[f(x_{k+1}) - f_{\text{low}} \mid \mathcal{T}_k\right] + c_\epsilon \;\leq\; f(x_k) - f_{\text{low}}. \tag{37}$$

As a result, we can apply Theorem 4.1 with $\alpha^k = f(x_k) - f_{\text{low}} \geq 0$, $\Sigma^k = \mathcal{T}_k$ and $\nu^k = c_\epsilon > 0$: we thus obtain that $\sum_{k=0}^{\infty} c_\epsilon < \infty$, which is obviously false. This implies that T_ϵ must be finite almost surely.

Consider now the sequence of random variables given by,

$$R_k = f(x_{\min(k,T_\epsilon)}) + \max\left(\min(k, T_\epsilon) - 1, 0\right) c_\epsilon.$$

Theorem 3.1 ensures that for any $k < T_\epsilon$, one has

$$\begin{aligned} \mathbb{E}[R_{k+1} \mid \mathcal{T}_k] &= \mathbb{E}[f(x_{k+1}) \mid \mathcal{T}_k] + k\, c_\epsilon \\ &\leq f(x_k) - c_\epsilon + k\, c_\epsilon \leq R_k \end{aligned}$$

and for $k \geq T_\epsilon$, trivially $R_{k+1} = R_k$, and thus R_k is a supermartingale. Moreover, since in $\mathcal{T}_k$, it holds that $T_\epsilon \geq k + 1$,

$$\begin{aligned} \mathbb{E}\left[|R_{k+1} - R_k| \mid \mathcal{T}_k\right] &= \mathbb{E}\left[|f(x_{k+1}) + (k+1)c_\epsilon - f(x_k) - kc_\epsilon| \mid \mathcal{T}_k\right] \\ &\leq \mathbb{E}\left[|f(x_{k+1}) - f(x_k)| + c_\epsilon \mid \mathcal{T}_k\right] \\ &\leq c_\epsilon + \max\left(c_\epsilon, f_{\max} - f_{\text{low}}\right) < \infty \end{aligned}$$

thus the expected increment between R_k and R_{k+1} is bounded. Since $T_\epsilon < \infty$ almost surely, we satisfy the assumptions of Theorem 4.2: it thus holds that $\mathbb{E}[R_{T_\epsilon}] \leq \mathbb{E}[R_0]$ which implies that,

$$f_{\text{low}} + (\mathbb{E}\left[T_\epsilon\right] - 1)c_\epsilon \leq \mathbb{E}\left[R_{T_\epsilon}\right] \leq \mathbb{E}[R_0] = f(x_0)$$

and thus,

$$\mathbb{E}\left[T_\epsilon\right] \;\leq \frac{(f(x_0) - f_{\text{low}})}{c_\epsilon} + 1.$$

Replacing c_ϵ by its definition gives the desired bound. ■

The result of Theorem 4.3 gives a worst-case complexity in the expected number of iterations until function stationarity. This does not provide a practical stopping criterion for the Algorithm, because only *model* stationarity can be tested for during an algorithmic run (note however that in the case of $\pi_k = p = 1$ both notions of stationarity are equivalent). However, by combining Theorem 4.3 and Lemma 3.1, we now that after up to $\mathcal{O}(\varrho(c\epsilon^{1/2}, p)\epsilon^{-3/2})$ iterations on average, if the model is accurate, then the corresponding iterate will be function stationary.

In our algorithm, we assume that accuracy is only guaranteed with a certain probability at every iteration. As a result, stopping after encountering an iterate that is model stationary only comes with a weak guarantee of returning a point that is function stationary. In developing an appropriate stopping criterion, we wish to avoid such "false positives". To this end, one possibility consists in requiring model stationarity to be satisfied for a certain number of successive iterations. This is the sense of the following proposition.

Proposition 4.1 *Under the assumptions of Theorem 4.3, suppose that Algorithm 1 reaches an iteration index $k + J$ such that for every $j \in \{k, k+1, \ldots, k+J\}$,*

$$\min\left\{\|g_j\|, \|g_j^+\|\right\} \leq \epsilon \quad \text{and} \quad \lambda_j \geq -\epsilon^{1/2}.$$

Suppose further that δ_g and δ_H satisfy (14)*, and that the sample sizes are selected independently of the current iterate. Then, with probability at least $1-(1-p)^{J+1}$, where p is the lower bound on p_k given by Assumption 3.4, one of the iterates $\{x_k, x_{k+1}, \ldots, x_{k+J}\}$ is $((1+\kappa_g)\epsilon, (1+\kappa_H)\epsilon^{1/2})$-function stationary.*

Proof. In this proof, we will use the notation $\mathbb{P}_k(\ldots) = \mathbb{P}(\cdot|\mathcal{F}_{k-1})$. Recall the event I_j from Definition 3.2. We also consider the random events

$$E \;=\; \left\{\text{One of the iterates in } \{x_{k+j}\}_{j=0..J} \text{ is } ((1+\kappa_g)\epsilon, (1+\kappa_H)\epsilon^{1/2})\text{-function stationary}\right\},$$

and

$$\forall j = 0, \ldots, J, \quad E_j \;=\; \left\{\text{The iterate } x_{k+j} \text{ is } (\epsilon, \epsilon^{1/2})\text{-model stationary}\right\}.$$

For every $j = 0, \ldots, J$, we have $E_j \in \mathcal{F}_{k+j}$ and $I_{k+j} \in \mathcal{F}_{k+j}$. Moreover, the event E_j and I_{k+j} are conditionally independent, i.e. for every $j = 0, \ldots, J$,

$$\mathbb{P}_k\left(E_j \cap I_{k+j} | \mathcal{F}_{k+j-1}\right) \;=\; \mathbb{P}_k\left(E_j | \mathcal{F}_{k+j-1}\right) \mathbb{P}_k\left(I_{k+j} | \mathcal{F}_{k+j-1}\right). \tag{38}$$

This conditional independence holds because $x_{k+j} \in \mathcal{F}_{k+j-1}$, and the model m_{k+j} is constructed independently of x_{k+j} by assumption.

The statement of the theorem corresponds to

$$\mathbb{P}_k\left(E | E_0, \ldots, E_J\right) \;\geq\; 1 - (1-p)^{J+1},$$

which is what we want to prove. By Lemma 3.1,

$$\mathbb{P}_k\left(E | E_0, \ldots, E_J\right) \;\geq\; \mathbb{P}_k\left(\bigcup_{0 \leq j \leq J} I_{k+j} \,\middle|\, E_0, \ldots, E_J\right) = 1 - \mathbb{P}_k\left(\bigcap_{0 \leq j \leq J} \bar{I}_{k+j} \,\middle|\, E_0, \ldots, E_J\right).$$

and it thus suffices to prove that

$$\mathbb{P}_k\left(\bigcap_{0\leq j\leq J}\bar{I}_{k+j}\middle|E_0,\ldots,E_J\right) \;\leq\; (1-p)^{J+1} \tag{39}$$

to obtain the desired result.

We now make use of the probabilistically accuracy property. For every $j=0,\ldots,J$, we have

$$\mathbb{P}_k\left(I_{k+j}|A\right)\geq p, \tag{40}$$

for any set of events A belonging to the σ-algebra $\mathcal{F}_{k+j-1}$ [16, Chapter 5]. In particular, for any $j\geq 1$, $\mathbb{P}_k\left(I_{k+j}|I_k,\ldots,I_{k+j-1},E_0,\ldots,E_j\right)\geq p$.

Returning to our target probability, we have:

$$\begin{aligned}
&\mathbb{P}_k\left(\bigcap_{0\leq j\leq J}\bar{I}_{k+j}\middle|E_0,\ldots,E_J\right)\\
&=\frac{\mathbb{P}_k\left(\left\{\bigcap_{0\leq j\leq J}\bar{I}_{k+j}\right\}\cap E_J\middle|E_0,\ldots,E_{J-1}\right)}{\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1}\right)}\\
&=\frac{\mathbb{P}_k\left(\bar{I}_J\cap E_J\middle|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)\mathbb{P}_k\left(\cap_{0\leq j\leq J-1}\bar{I}_{k+j}\middle|\;E_0,\ldots,E_{J-1}\right)}{\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1}\right)}\\
&=\mathbb{P}_k\left(\bar{I}_J\middle|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)\mathbb{P}_k\left(\cap_{0\leq j\leq J-1}\bar{I}_{k+j}\middle|\;E_0,\ldots,E_{J-1}\right)\\
&\times\frac{\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)}{\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1}\right)}\\
&\leq\mathbb{P}_k\left(\bar{I}_J\middle|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)\mathbb{P}_k\left(\cap_{0\leq j\leq J-1}\bar{I}_{k+j}\middle|\;E_0,\ldots,E_{J-1}\right).
\end{aligned}$$

where the last equality comes from (38), and the final inequality uses the fact that the events $E_0,\ldots,E_{J-1}$ and $I_k,\ldots,I_{k+J-1}$ are pairwise independent, thus

$$\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)\;=\;\mathbb{P}_k\left(E_J|E_0,\ldots,E_{J-1}\right).$$

Using (40), we then have that

$$\mathbb{P}_k\left(\bar{I}_J\middle|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)=1-\mathbb{P}_k\left(I_J|E_0,\ldots,E_{J-1},I_k,\ldots,I_{k+J-1}\right)\leq 1-p.$$

Thus,

$$\mathbb{P}_k\left(\bigcap_{0\leq j\leq J}\bar{I}_{k+j}\middle|E_0,\ldots,E_J\right)\leq(1-p)\mathbb{P}_k\left(\bigcap_{0\leq j\leq J-1}\bar{I}_{k+j}\middle|E_0,\ldots,E_{J-1}\right). \tag{41}$$

By a recursive argument on the right-hand side of (41), we thus arrive at (39), which yields the desired conclusion.

■

The result of Proposition 4.1 is only of interest if we are guaranteed that such a sequence of model stationary iterates can occur in a bounded number of iterations. This property can be ensured by rejecting a step if it is deemed that the new iterate is not promising in terms of decrease. This is the sense of the following assumption.

Assumption 4.1 *In Algorithm 1, Step 7 is replaced by:*
7'. If $\min\{\|g_k\|, \|g_k^+\|\} < \epsilon$ and $\lambda_k > -\epsilon^{1/2}$, set $x_{k+1} = x_k$, otherwise set $x_{k+1} = x_k + \alpha_k d_k$.
Moreover, the sample size is selected independently of the current iterate.

This algorithm change allows us to measure stationarity at a given iterate based on a series of samples. Under a slightly stronger set of assumptions, Proposition 4.2 then guarantees that sequences of model stationary points will occur in expectation.

Proposition 4.2 *Let Assumptions 3.1, 3.2, 3.3, 3.4 and 4.1 hold, where δ satisfies (25) with $\kappa_g, \kappa_H \in (0,1)$, and $p_k = p$ $\forall k$. For a given $J \in \mathbb{N}$, define $T^m_{\epsilon,J}$ as the first iteration index of Algorithm 1 for which*

$$\min\{\|g_k\|, \|g_k^+\|\} < \epsilon \quad \textit{and} \quad \lambda_k > -\epsilon^{1/2}, \quad \forall k \in \{T^m_{\epsilon,J}, T^m_{\epsilon,J}+1, ..., T^m_{\epsilon,J}+J\}. \tag{42}$$

Suppose finally that for every index k, the sample size π_k satisfies

$$\forall k, \quad \pi_k \geq \varrho(c\hat{\epsilon}^{1/2}, p), \tag{43}$$

where $\hat{\epsilon} = \min\{\frac{1-\kappa_g}{1+\kappa_g}, \frac{(1-\kappa_H)^2}{(1+\kappa_H)^2}\}\epsilon$. (Note that $\varrho(c\hat{\epsilon}^{1/2}, p) \geq \varrho(c\epsilon^{1/2}, p)$.)
Then, $T^m_{\epsilon,J} < \infty$ almost surely, and

$$\mathbb{E}[T^m_{\epsilon,J}] \leq \begin{cases} \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\epsilon^{-3/2} + J + 1 & \textit{if } \pi_k = 1 \ \forall k; \\ p^{-(J+1)}\left[\frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\,\varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\,\hat{\epsilon}^{-3/2} + J + 1\right] & \textit{otherwise.} \end{cases} \tag{44}$$

Proof. As in Theorem 4.3, $T^m_{\epsilon,J}$ clearly is a stopping time. Moreover, if $\pi_k = 1$ for all k, then $T^m_{\epsilon,J} = T_\epsilon$ for every J, where T_ϵ is the stopping time defined in Theorem 4.3, and therefore the result holds. In what follows, we thus focus on the remaining case.

Consider an iterate k such that x_k is $(\hat{\epsilon}, \hat{\epsilon}^{1/2})$-function stationary and the model m_k is accurate. From the definition of $\hat{\epsilon}$, such an iterate is also $((1-\kappa_g)\epsilon, (1-\kappa_H)\epsilon^{1/2})$ function stationary and the model m_k is accurate. Then, by a reasoning similar to that of the proof of Lemma 3.1, we can show that x_k is $(\epsilon, \epsilon^{1/2})$-model stationary. As a result, if $T^m_{\epsilon,J} > k$, one of the models $m_k, m_{k+1}, \ldots, m_{k+J}$ must be inaccurate, which happens with probability $1 - p^{J+1}$.

Let $T_{\epsilon,J}$ be the first iteration index for which the iterate is a $(\hat{\epsilon}, \hat{\epsilon}^{1/2})$ function stationary point and satisfies (42). Clearly $T^m_{\epsilon,J} \leq T_{\epsilon,J}$ (for all realizations of these two stopping times), and it thus suffices to bound $T_{\epsilon,J}$ in expectation. By applying Theorem 4.3 (with ϵ in the theorem's statement replaced by $\hat{\epsilon}$), one can see that there must exist an infinite number of $(\hat{\epsilon}, \hat{\epsilon}^{1/2})$-function stationary points in expectation. More precisely, letting $\{T^{(i)}_{\hat{\epsilon}}\}_{i=1,\ldots}$ be the corresponding stopping times indicating the iteration indices of these points and using Theorem 4.3 (excluding the case $\pi_k = 1$ $\forall k$), we have

$$\begin{aligned} \mathbb{E}\left[T^{(1)}_{\hat{\epsilon}}\right] = \mathbb{E}\left[T_{\hat{\epsilon}}\right] &\leq \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\,\varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\,\hat{\epsilon}^{-3/2} + 1, \\ \forall i \geq 1, \quad \mathbb{E}\left[T^{(i+1)}_{\hat{\epsilon}} - T^{(i)}_{\hat{\epsilon}}\right] &\leq \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\,\varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\,\hat{\epsilon}^{-3/2} + 1. \end{aligned}$$

Consider now the subsequence $\{T_{\hat{\epsilon}}^{(i_\ell)}\}_{\ell=1,\dots}$ such that all stopping times are at least J iterations from each other, i.e., for every $\ell \geq 1$, we have $T_{\hat{\epsilon}}^{(i_{\ell+1})} - T_{\hat{\epsilon}}^{(i_\ell)} \geq J$. For such a sequence, we get

$$\begin{aligned} \forall \ell \geq 1, \quad \mathbb{E}\left[T_{\hat{\epsilon}}^{(i_{\ell+1})} - T_{\hat{\epsilon}}^{(i_\ell)}\right] &\leq \frac{(f(x_0) - f_{\text{low}})}{\hat{c}}\, \varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\, \hat{\epsilon}^{-3/2} + J + 1 \triangleq K(\epsilon, J) \\ \mathbb{E}\left[T_{\hat{\epsilon}}^{(i_1)}\right] = \mathbb{E}\left[T_{\hat{\epsilon}}\right] &\leq \frac{(f(x_0) - f_{\text{low}})}{\hat{c}}\, \varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\, \hat{\epsilon}^{-3/2} + 1 \leq K(\epsilon, J). \end{aligned}$$

For every $\ell \geq 1$, we define the event

$$B_\ell \;=\; \bigcap_{j=0}^{J} I_{T_{\hat{\epsilon}}^{(i_\ell)}+j} \;=\; \bigcap_{j=0}^{J} \left\{ m_{T_{\hat{\epsilon}}^{(i_\ell)}+j} \text{ is accurate} \right\}.$$

By assumption, the models are generated independently of the current iterate, and for every k, $\mathbb{P}(I_k|\mathcal{F}_{k-1}) = p$. By the same recursive reasoning as in the proof of Proposition 4.1, we have that $\mathbb{P}\left(B_\ell|\mathcal{F}_{T_{\hat{\epsilon}}^{(i_\ell)}-1}\right) = p^{J+1}$. Moreover, by definition of the sequence $\{T_{\hat{\epsilon}}^{(i_\ell)}\}$, two stopping times in that sequence correspond to two iteration indices distant of at least $J+1$. Therefore, they also correspond to two separate sequences of $(J+1)$ models that are generated in an independent fashion. We can thus consider $\{B_\ell\}$ to be an independent sequence of Bernoulli trials. Therefore, the variable G representing the number of runs of B_ℓ until success follows a geometric distribution with an expectation less than $\frac{1}{p^{J+1}} = p^{-(J+1)} < \infty$. On the other hand, $T_{\epsilon,J}$ is less than the first element of $\{T_{\hat{\epsilon}}^{(i_\ell)}\}$ for which B_ℓ happens, and thus $T_{\epsilon,J} \leq T_{\hat{\epsilon}}^{(i_G)}$. To conclude the proof, we define

$$S_G = T_{\hat{\epsilon}}^{(i_G)}, \qquad X_1 = T_{\hat{\epsilon}}^{(i_1)} = T_{\hat{\epsilon}}^{1}, \quad X_\ell = T_{\hat{\epsilon}}^{(i_\ell)} - T_{\hat{\epsilon}}^{(i_{\ell-1})} \quad \forall \ell \geq 2.$$

From the proof of Wald's equation [16, Theorem 4.1.5] (more precisely, from the third equation appearing in that proof), one has

$$\mathbb{E}\left[S_G\right] = \sum_{\ell=1}^{\infty} \mathbb{E}\left[X_\ell\right] \mathbb{P}(G \geq \ell).$$

Since $\mathbb{E}\left[X_\ell\right] \leq K(\epsilon, J)$, one arrives at

$$\mathbb{E}\left[T_{\epsilon,J}^m\right] \leq \mathbb{E}\left[T_{\epsilon,J}\right] \leq \mathbb{E}\left[T_{\hat{\epsilon}}^{(i_G)}\right] \leq K(\epsilon, J) \sum_{\ell=1}^{\infty} \mathbb{P}(G \geq \ell) = K(\epsilon, J)\mathbb{E}\left[G\right],$$

which is the desired result. ■

With the result of Proposition 4.2, we are guaranteed that there will exist consecutive iterations satisfying model stationarity in expectation: checking for those thus represents a valid stopping criterion in practice. If an estimate of the probability p is known, one can even choose J to guarantee that the probability of computing a stationary iterate is sufficiently high.

Evaluation complexity: As a final note to this section, we observe that the number of evaluations of f_i and their derivatives is bounded in expectation, thanks to the result of Theorem 4.3. Note that we must account for the additional objective evaluations induced by the line-search process (see [29, Theorem 8] for details).

Corollary 4.1 *Under the assumptions of Proposition 4.2, assuming that the sample size condition* (27) *is satisfied with* $\pi_k \in \{\varrho(c\hat{\epsilon}^{1/2}, p), 1\}$ *for every* k*, the expected number of evaluations of* ∇f_i *and* $\nabla^2 f_i$ *are respectively bounded above by*

$$\begin{cases} \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\epsilon^{-3/2} + 1 & \text{if } \pi_k = 1 \ \forall k; \\ p^{-(J+1)}\left[\frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\ \varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\,\hat{\epsilon}^{-3/2} + J + 1\right] & \text{otherwise,} \end{cases} \tag{45}$$

while the expected number of function evaluations is bounded above by

$$(1+\max\{j_{nc}, j_n, j_{rn}\}) \times \begin{cases} \frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\epsilon^{-3/2} + 1 & \text{if } \pi_k = 1 \ \forall k; \\ p^{-(J+1)}\left[\frac{(f(x_0)-f_{\text{low}})}{\hat{c}}\ \varrho(c\hat{\epsilon}^{1/2}, p)^{-1}\,\hat{\epsilon}^{-3/2} + J + 1\right] & \text{otherwise.} \end{cases} \tag{46}$$

The bounds (45) and (46) match their deterministic counterparts (see [29]) in the worst case.

4.2 Illustration of sample and evaluation complexity for uniform sampling

The two main requirements made in the analysis of the previous section are related to the function value accuracy δ_f and the sample size π_k. Specifically, for a given tolerance ϵ, we required in Theorem 4.3 that:

1. $\delta_f \leq \frac{\eta}{24}c^3\epsilon^{3/2},\ \delta_g \leq \kappa_g\epsilon,\ \delta_H \leq \kappa_H\epsilon^{1/2}$.
2. $\pi_k \geq \varrho(c\epsilon^{1/2}, p) = \frac{(1-p)\left(U_g \max\{U_H, U_g^{1/2}\} + \frac{L}{2}\max\{U_h^2, U_g, \epsilon^{-1}U_g^2\}\right)}{(1-p)\left(U_g \max\{U_H, U_g^{1/2}\} + \frac{L}{2}\max\{U_h^2, U_g\}\right) + p\frac{\eta}{24}c^3\epsilon^{3/2}}$.

In this section, we provide estimates of the minimum number of samples necessary to achieve those two conditions in the case of a uniform sampling strategy. To facilitate the exposure, we discard the case $p = 1$ and focus on $p \in (0, 1)$. Note that a similar analysis also holds for the requirements of Proposition 4.2.

For the rest of the section, we suppose that the set $\mathcal{S}_k$ is formed of $n\bar{\pi}$ indices chosen uniformly at random with replacement. where $\bar{\pi}$ is independent of k. That is, for every $i \in \mathcal{S}_k$ and every $j = 1, \ldots, N$, we have $\mathbb{P}(i = j) = \frac{1}{N}$. This case has been well studied in the case of subsampled Hessian [31] and subsampled gradient [27]. The lemmas below summarize the results.

Lemma 4.1 *Under Assumptions 3.2 and 3.3, consider an iterate* x_k *of Algorithm 1. For any* $\varrho \in (0, 1)$*, if the sample set* $\mathcal{S}_k$ *is chosen to be of size*

$$\bar{\pi} \geq \frac{1}{N}\frac{16L^2}{\delta_H}\ln\left(\tfrac{2N}{1-p}\right), \tag{47}$$

then

$$\mathbb{P}\left(\left\|H(x_k; \mathcal{S}_k) - \nabla^2 f(x_k)\right\| \leq \delta_H \middle| \mathcal{F}_{k-1}\right) \geq p.$$

Proof. See [31, Lemma 4]; note that here we are using L_i (Lipschitz constant of ∇f_i) as a bound on $\|\nabla^2 f_i(x_k)\|$, and that we are providing a bound on the sampling fraction $\bar{\pi} = \frac{|\mathcal{S}_k|}{N}$. ■

By the same reasoning as for Lemma 4.1, but in one dimension, we can readily provide a sample size bound for obtaining accurate function values. To this end, we define

$$f_{\text{up}} \geq \max_k \max_{i=1,..,N} f_i(x_k). \tag{48}$$

Note that there exists such a bound when Assumption 3.3 is satisfied, and specific structure of the problem could allow for it to available (in the case of a loss function, for instance, one can often choose $f_{\text{up}} = 1$).

Lemma 4.2 *Under Assumptions 3.2 and 3.3, consider an iterate x_k of Algorithm 1. For any $p \in (0,1)$, if the sample set $\mathcal{S}_k$ is chosen to be of size*

$$\bar{\pi} \geq \frac{1}{N} \frac{16 f_{\text{up}}^2}{\delta_f} \ln\left(\tfrac{2}{1-p}\right), \tag{49}$$

then

$$\mathbb{P}\left(|m(x_k;\mathcal{S}_k) - f(x_k)| \leq \delta_f | \mathcal{F}_{k-1}\right) \geq p.$$

Proof. The proof follows that of [31, Lemma 4.1] by considering $m(x_k;\mathcal{S}_k)$ and $f(x_k)$ as one-dimensional matrices. ■

Lemma 4.3 *Under Assumptions 3.2 and 3.3, consider an iterate x_k of Algorithm 1. For any $\varrho \in (0,1)$, if the sample set $\mathcal{S}_k$ is chosen to be of size*

$$\bar{\pi} \geq \frac{1}{N} \frac{U_g^2}{\delta_g^2} \left[1 + \sqrt{8 \ln\left(\tfrac{1}{1-p}\right)}\right]^2, \tag{50}$$

then

$$\mathbb{P}\left(\|g(x_k;\mathcal{S}_k) - \nabla f(x_k)\| \leq \delta_g | \mathcal{F}_{k-1}\right) \geq p.$$

Proof. See [27, Lemma 2]. ■

By combing those three lemmas, we obtain the following sample size condition.

Theorem 4.4 *Let Assumptions 3.1, 3.2 and 3.3 hold. Suppose that the sample fractions π_k of Algorithm 1 are chosen to satisfy $\pi_k \geq \pi(\epsilon)$ for every k, where*

$$\pi(\epsilon) := \frac{1}{N} \max\left\{ N \ \varrho(c\epsilon^{1/2}, p), \frac{344 f_{\text{up}}^2}{\eta c^3 \epsilon^{3/2}} \ln\left(\tfrac{2}{1-p}\right), \frac{U_g^2}{\kappa_g^2 \epsilon^2}\left[1 + \sqrt{8 \ln\left(\tfrac{1}{1-p}\right)}\right], \frac{16 L^2}{\kappa_H \epsilon^{1/2}} \ln\left(\tfrac{2N}{1-p}\right) \right\}.$$

Then, the model sequence is p-probabilistically $(\delta_f, \delta_g, \delta_H)$-accurate with $\delta_f = \frac{\eta}{24} c^3 \epsilon^{3/2}$, $\delta_g = \kappa_g \epsilon$ and $\delta_H = \kappa_H \epsilon^{1/2}$. Moreover, the results from Section 4.1 hold.

We observe that explicit computation of these bounds would require estimating L, L_H, U_H, and U_g. If orders of magnitude for the aforementioned quantities are available, they can be used for choosing the sample size.

4.3 Comparison with other sample complexities

By looking at several existing sample complexity results in the literature, we can position our method within the existing landscape of results, and get insight about the cost of second-order requirements, as well as that of using inexact function values.

When applied to nonconvex problems, a standard stochastic gradient approach with fixed step size has a complexity in $\mathcal{O}(\epsilon^{-4})$ (both in terms of iterations and gradient evaluations) for reaching approximate first-order optimality [17]. Modified SGD methods that take curvature information can significantly improve over that bound, one recent example being the Natasha 2 method that requires $\mathcal{O}(\epsilon^{-3.25})$ iterations/gradient evaluations for reaching $\|\nabla f(x_k)\| \leq \epsilon$ and $\nabla^2 f(x_k) \succeq -\epsilon^{1/4} I$ [2]. Note that those guarantees are on the true derivatives, and not their expected value/estimates. These orders hold in high probability, and thus are not informative in the event of a poor approximation at one step of the procedure. Our results are able to cover such a situation. Note that we can also turn our results to high-probability rates, following for instance an argument similar to [18].

It is also interesting to compare our complexity orders with those obtained by first-order methods in stochastic optimization. Stochastic trust-region methods [13, 21] typically require $\mathcal{O}\left(\Delta_k^{-4}\right)$ samples per iteration, where Δ_k is the trust-region radius and serves as an approximation of the norm of the gradient. Similarly, the line-search algorithm of Paquette and Scheinberg [25] guarantees sufficient accuracy in the function values if the sample size is of order $\mathcal{O}\left(\alpha_k^{-2}\|g_k\|^{-4}\right)$ (we use our notations for consistency), which for our method roughly corresponds to $\mathcal{O}\left((\alpha_k\|d_k\|)^{-2}\right)$. Our sample complexity is of order $\mathcal{O}\left((\alpha_k\|d_k\|)^{-3}\right)$ but possesses second-order convergence rates. One could thus conjecture that endowing the algorithm of [25] with second-order guarantees in expectation would increase the sample complexity from $\mathcal{O}\left((\alpha_k\|d_k\|)^{-2}\right)$ to $\mathcal{O}\left((\alpha_k\|d_k\|)^{-3}\right)$.

Finally, we discuss sample bounds for Newton-based methods in a subsampling context. In [31], it was shown that the desired complexity rate of $\mathcal{O}(\epsilon^{-3/2})$ is achieved with probability $1-p$, if the proportion of samples taken from the sum of samples satisfies

$$|\mathcal{S}| \geq \frac{16U_H^2}{\epsilon^2} \ln\left(\frac{2N}{p}\right)$$

and in [23] we have a similar result where $\mathcal{S}$ must satisfy

$$|\mathcal{S}| \geq \frac{16L^2}{\epsilon^2} \ln\left(\frac{2N}{p}\right).$$

In both cases, the desired samples are proportional to a constant measuring the size of the problem data and inversely proportional to the desired tolerance squared, as in our case. In [33], high probability results were derived with the accuracy of the function gradient and Hessian estimates δ_g and δ_H being bounded by ϵ. In part because of the accuracy requirements on the gradient, the resulting sampling bounds are $\mathcal{O}(\epsilon^{-2})$. In our setting, we require the function accuracy to be of order of order $\epsilon^{3/2}$, where ϵ is the gradient tolerance, yet the dependencies on ϵ suggested by Lemma 3.1 and 4.4 are similar to those in the literature [31, 33]: they are even identical in terms of gradient and Hessian sampling costs. As mentioned above, the sampling rate conditions are closely related to the norm of the steps. This connection was used to define probabilistic models in trust-region methods [3]. We believe that a version of our algorithm that would assume access to exact function values but subsampling derivatives would have the

same sample complexity than the aforementioned methods [31, 33]: our analysis and discussion regarding stationarity would remain relevant in this context, because of our use of inexact derivatives.

5 Numerical Illustration

In this section, as a proof of concept, we will evaluate the performance of our proposed algorithm in comparison with classical gradient-based methods. To this end, we consider a binary classification problem where one tries to minimize a non-linear least squares loss function. More precisely, given a training set $\{z_i, y_i\}_{i=1}^N$ where $z_i \in \mathbb{R}^n$ and $y_i \in \{0, 1\}$, we consider the following empirical risk minimization problem,

$$\min_{x \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^{N} (y_i - \Phi(z_i^\top x))^2, \tag{51}$$

where Φ is the sigmoid function, i.e., $\Phi(w) = \frac{1}{1+\exp(-w)}$ for all $w \in \mathbb{R}$. Our training data comes from the `IJCNN1` dataset, itself obtained from the LIBSVM library [12]: for this dataset, we have $N = 49990$ and $n = 22$.

The basic parameters in Algorithm 1 are set to $\epsilon = 10^{-5}$, $\eta = 10^{-2}$, and $\theta = 0.9$. Our tests include three variants of Algorithm 1 identified as follows:

- `ALAS-ALGO (full)`: Algorithm 1 using exact function, gradient and Hessian (i.e., using the entire dataset, $\mathcal{S} = \{1, \ldots, N\}$).
- `ALAS-ALGO (5%)`: Algorithm 1 with a uniform subsampling to generate $\mathcal{S}$ and using 5% of the entire dataset.
- `ALAS-ALGO (1%)`: Algorithm 1 with a uniform subsampling to generate $\mathcal{S}$ and using 1% of the entire dataset.

Similarly, our experiments include three variants of a basic implementation of the stochastic gradient descent method, with the same sampling strategies as the `ALAS-ALGO` schemes. Those variants are called `SGD-ALGO (full)` (equivalent to deterministic gradient descent), `SGD-ALGO (5%)` and `SGD-ALGO (1%)`, and all three variants use a step size of 0.1. As a stopping criterion for all the tested solvers, we require $(\epsilon, \epsilon^{1/2})$-model stationarity with $\epsilon = 10^{-5}$. When subsampling is used, $(\epsilon, \epsilon^{1/2})$-model stationarity must be satisfied for J successive iterations, where we chose J to be 1 epoch (1 epoch of a solver that uses s% of the dataset is equivalent to running $\frac{100}{s}$ iterations of this method). This choice was made after testing a number of values, and we believe that it serves our illustrative purpose by exploiting the link between model stationarity and true objective stationarity studied in Section 4.1.

For all tested solvers, we impose a maximum number of 1600 epochs; in our numerical tests, this maximum number of epochs ensured that at least `ALAS-ALGO (full)` reached the $(\epsilon, \epsilon^{1/2})$-stationarity (note that in this case, as the full dataset is used, 1 epoch represents one iteration of the algorithm). To account for the randomness in the subsampling process, we performed multiple runs started from different random points. Our runs all yielded similar results, therefore we describe our findings based upon results from a single representative run.

The obtained results are depicted in Figures 1 and 2. The true function value and exact norm gradients are plotted in Figure 1, while Figure 2 shows the exact function values (respectively, exact gradient norm) and the model estimates (respectively, the sampled gradient norm) for `ALAS-ALGO` using 1% and 5% of the dataset. Overall, several patterns are fairly consistent. First, `ALAS-ALGO` takes much fewer iterations than all the variants of `SGD-ALGO` to decrease the objective value and gradient norm, showing the effect of using second-order information. Admittedly the iteration cost of `ALAS-ALGO` is higher than SGD, yet the sampling procedure is performed with the same batch size for both: from that metric, `ALAS-ALGO` yields a better error for the same order of sampling, and requires less passes through the data. We note that both kinds of subsampling methods start plateauing after an initial, fast convergence phase. Our second observation is that for our subsampled methods, convergence is very fast in the beginning of the run in expectation: the methods then enter a regime in which the noise induced by the subsampling process slows down the convergence, as is also the case for the SGD methods. We observe however that the good initial behavior of `ALAS-ALGO` allows the algorithm to reach better function values for the same number of epochs.

6 Conclusion

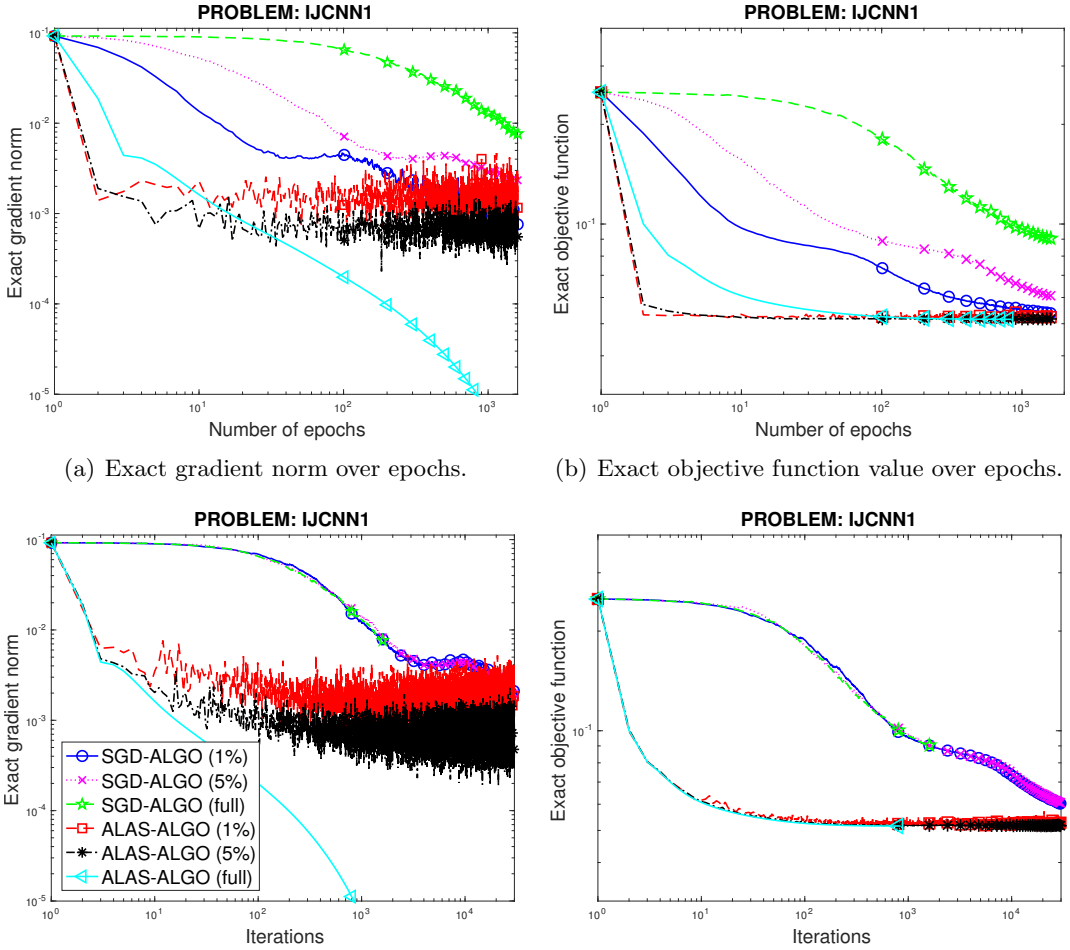
In this paper, we presented a line-search method for stochastic optimization, wherein the Hessian, gradient, and function values must be estimated through subsampling and cannot be obtained exactly. Using probabilistically accurate models, we derived a complexity result on the expected number of iterations until an approximate measure of stationary is reached for the current model. This result in expectation is complementary to those holding with high probability, i.e., if an accurate sample is taken at every iteration. In our setting, we are able to obtain convergence in expectation at a rate which is commensurate with the state of the art, regardless of the presence of outliers in sample estimates resulting in poor steps. We also proposed a practical strategy to assess whether the current iterate is close to a sample point for the original objective, that does not require the computation of the full function. Our preliminary numerical results showed the potential of the proposed approach on a standard dataset.

We believe the results of the paper complement the existing literature in a contributive way, and will encourage further study into the design of algorithms appropriate for large scale optimization that are robust with respective to poor sample approximations. The primarily theoretical contributions of this paper will be informative in regards to the design and analysis of stochastic algorithms. In particular, our approach could be helpful in generalizing other line-search techniques to the context of subsampled function values.

To the best of the authors' knowledge, currently the closest implementation of the procedure described in this paper is given in [20], using proprietary data and thus limited experiments. We are currently working on an extensive numerical study, based on exploiting the practical features of [20] (e.g., inexact matrix operations), on a wider variety of problems implementing training of complex deep neural net architectures. We believe that this ongoing work will further advance the state of the art in the study of stochastic algorithms for finite sum optimization.

Acknowledgments

We are indebted to Courtney Paquette and Katya Scheinberg for raising an issue with the first version of this paper, that lead to significant improvement of its results.



(a) Exact gradient norm over epochs.

(b) Exact objective function value over epochs.

(c) Exact gradient norm over iterations.

(d) Exact objective function value over iterations.

Figure 1: Obtained results on `IJCNN1` for binary classification. Logarithmic scales are used for the x-axis and y-axis. The curve markers are placed every 10^2 epochs for Figures (1(a)) and (1(b)), and every 800 iterations for Figures (1(c)) and (1(d)).

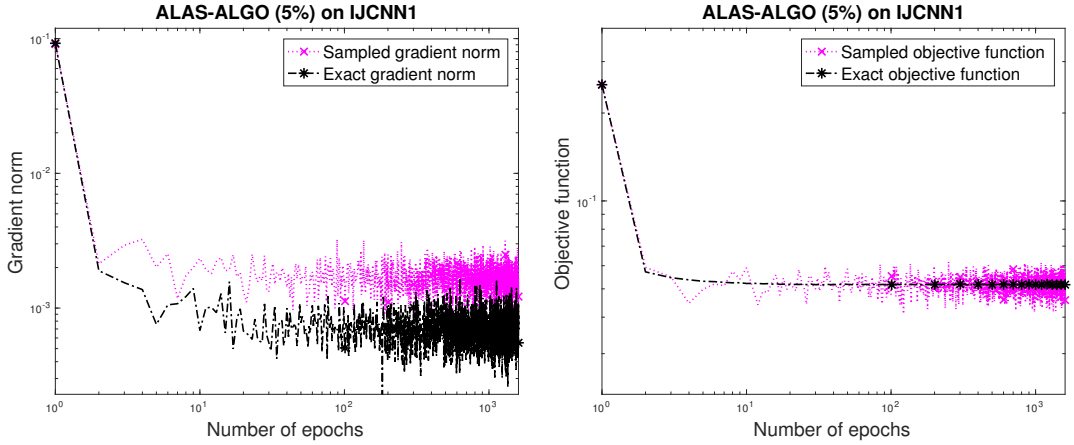
References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1195–1199, 2017.

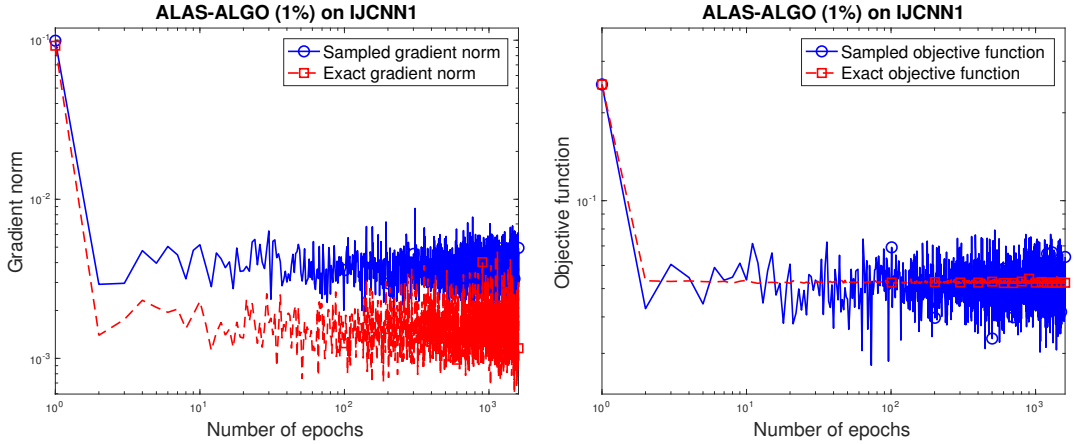
[2] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than SGD. *arXiv preprint arXiv:1708.08694*, 2017.

[3] Afonso S Bandeira, Katya Scheinberg, and Luis N Vicente. Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.*, 24:1238–1264, 2014.

[4] El Houcine Bergou, Youssef Diouane, Vyacheslav Kungurtsev, and Clément W Royer. A stochastic Levenberg-Marquardt method using random models with application to data assimilation. *arXiv preprint arXiv:1807.2176*, 2018.



(a) $\nabla f(x_k)$ versus $\nabla m_k(x_k)$. (b) $f(x_k)$ versus $m_k(x_k)$.



(c) $\nabla f(x_k)$ versus $\nabla m_k(x_k)$. (d) $f(x_k)$ versus $m_k(x_k)$.

Figure 2: Obtained results on `IJCNN1` for binary classification using `ALAS-ALGO` solvers. Logarithmic scales are used for the x-axis and y-axis. The curve markers are placed every 10^2 epochs for Figures (2(a)) and (2(b)), and every 800 iterations for Figures (2(c)) and (2(d)).

[5] El Houcine Bergou, Serge Gratton, and Luis N Vicente. Levenberg-Marquardt methods based on probabilistic gradient models and inexact subproblem solution, with application to data assimilation. *SIAM/ASA J. Uncertain. Quantif.*, 4:924–951, 2016.

[6] Jose Blanchet, Coralia Cartis, Matt Menickelly, and Katya Scheinberg. Convergence rate analysis of a stochastic trust region method for nonconvex optimization. *arXiv preprint arXiv:1609.07428*, 2016 (Version 2).

[7] Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching L-BFGS method for machine learning. *arXiv preprint arXiv:1802.05374*, 2018.

[8] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60:223–311, 2018.

[9] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *SIAM J. Optim.*, 28:1751–1772, 2018.

[10] Coralia Cartis, Nicholas I M Gould, and Philippe L Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130:295–319, 2011.

[11] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Program.*, 169:337–375, 2018.

[12] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Syst. Tech.*, 2:1–27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[13] Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. *Math. Program.*, 169:447–487, 2018.

[14] Frank E Curtis, Katya Scheinberg, and Rui Shi. A stochastic trust-region algorithm. Technical Report 17T-017, COR@L Laboratory, Department of ISE, Lehigh University, 2017.

[15] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2933–2941, 2014.

[16] Rick Durrett. *Probability: Theory and Examples*. Camb. Ser. Stat. Prob. Math. Cambridge University Press, Cambridge, fourth edition, 2010.

[17] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23:2341–2368, 2013.

[18] Serge Gratton, Clément W Royer, Luis N Vicente, and Zaikun Zhang. Complexity and global rates of trust-region methods based on probabilistic models. *IMA J. Numer. Anal.*, 38:1579–1597, 2018.

[19] Jonas Moritz Kohler and Aurélien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In *International Conference on Machine Learning (ICML)*, pages 1895–1904, 2017.

[20] Vyacheslav Kungurtsev and Tomas Pevny. Algorithms for solving optimization problems arising from deep neural net models: smooth problems. *Cisco-CTU WP5 Technical Report*, 2016. Originally proprietary industrial report. Currently available online at https://arxiv.org/abs/1807.00172.

[21] Jeffrey Larson and Stephen C Billups. Stochastic derivative-free optimization using a trust region framework. *Comput. Optim. Appl.*, 64:619–645, 2016.

[22] Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid saddle points. *arXiv preprint arXiv:1710.07406*, 2017.

[23] Mingrui Liu and Tianbao Yang. On noisy negative curvature descent: Competing with gradient descent for faster non-convex optimization. *arXiv preprint arXiv:1709.08571*, 2017.

[24] Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. *J. Mach. Learn. Res.*, 18:1–59, 2017.

[25] Courtney Paquette and Katya Scheinberg. A stochastic line search method with convergence rate analysis. *arXiv preprint arXiv:1807.07994*, 2018.

[26] Herbert Robbins and David Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pages 233–257. Elsevier, 1971.

[27] Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods I: Globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016.

[28] Sheldon M Ross. *Stochastic processes*. Wiley, New York, 1996.

[29] Clément W Royer and Stephen J Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM J. Optim.*, 28:1448–1477, 2018.

[30] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *arXiv preprint arXiv:1711.02838*, 2017.

[31] Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Newton-type methods for non-convex optimization under inexact hessian information. *arXiv preprint arXiv:1708.07164*, 2017.

[32] Yi Xu, Rong Jin, and Tianbao Yang. First-order stochastic algorithms for escaping from saddle points in almost linear time. *arXiv preprint arXiv:1711.01944*, 2018 (version 3).

[33] Zhewei Yao, Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Inexact non-convex Newton-type methods. *arXiv preprint arXiv:1802.06925*, 2018.