A stochastic cubic regularisation method with inexact function evaluations and random derivatives for finite sum minimisation

Bellavia, Stefania; Gurioli, Gianmarco; Morini, Benedetta; Toint, Philippe L.

Publication date:
2020

Document Version
Peer reviewed version

A stochastic cubic regularisation method with inexact function evaluations and random derivatives for finite sum minimisation

Stefania Bellavia [*1] **Gianmarco Gurioli** [*2] **Benedetta Morini** [*1] **Philippe L. Toint** [*3]

Abstract

This paper focuses on an Adaptive Cubic Regularisation (ARC) method for approximating a second-order critical point of a finite sum minimisation problem. The variant presented belongs to the framework of (Bellavia et al., 2020b): it employs random models with accuracy guaranteed with a sufficiently large prefixed probability and deterministic inexact function evaluations within a prescribed level of accuracy. Without assuming unbiased estimators, the expected number of iterations is $\mathcal{O}\big(\epsilon_1^{-3/2}\big)$ or $\mathcal{O}\big(\max[\epsilon_1^{-3/2}, \epsilon_2^{-3}]\big)$ when searching for a first- or second-order critical point, respectively, where ϵ_j, $j \in \{1, 2\}$, is the jth-order tolerance. These results match the worst-case optimal complexity for the deterministic counterpart of the method.

1. Introduction

We consider an ARC method to compute an approximate q-th order, $q \in \{1, 2\}$, local minimum of the finite sum minimisation problem

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

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, ..., N\}$. The wide range of methods used in literature to solve (1) can be classified as first-order methods, requiring only the gradient of the objective f and second-order procedures, where the Hessian is also needed. Although first-order schemes are generally characterised by a simple and low-cost iteration, their performance can be seriously hindered by ill-conditioning and their success is highly dependent on the fine-tuning of hyper-parameters. In addition, the objective function in (1) can be nonconvex, with a variety of local minima and/or saddle points. For this reason, second-order strategies using curvature information have been used recently as an instrument for escaping more easily from saddle points (Bottou et al., 2018; Cartis et al., 2012a; Xu et al., 2020; Berahas et al., 2020). Clearly, the per-iteration cost is higher than for first-order methods, since second-order derivatives information is needed. By contrast, second-order methods have been shown to be significantly more resilient to ill-conditioned and badly-scaled problems, less sensitive to the choice of hyper-parameters and tuning (Berahas et al., 2020; Xu et al., 2020). We thus focus on second-order methods, building on the basic ARC approach because of its optimal complexity (Bellavia et al., 2019). In this framework, a first- or second-order stationary point can be achieved in at most $\mathcal{O}\big(\epsilon_1^{-3/2}\big)$ or $\mathcal{O}\big(\max[\epsilon_1^{-3/2}, \epsilon_2^{-3}]\big)$ iterations, respectively (Birgin et al., 2017; Cartis et al., 2012a;b; 2011b). Allowing inexactness in the function and/or derivative evaluations of ARC methods, while preserving convergence properties and optimal complexity, has been a challenge in recent years (Bellavia et al., 2020a; Chen et al., 2018b; Xu et al., 2019; Cartis & Scheinberg, 2018; Kohler & Lucchi, 2017; Xu et al., 2020; Yao et al., 2018; Zhou et al., 2019). A first approach is to impose suitable accuracy requirements that f and the derivatives have to deterministically fulfill at each iteration. But in machine learning applications function and derivatives are generally approximated by using uniformly randomly selecting subsets of terms in the sums and the accuracy levels can be satisfied only within a certain probability (Bellavia et al., 2019; 2020a; Chen et al., 2018b; Xu et al., 2019; Yao et al., 2018; Berahas et al., 2020; Cartis & Scheinberg, 2018). This suggests a stochastic analysis of the expected worst-case number of iterations needed to find a first- or second-order critical point (Cartis & Scheinberg, 2018; Zhou et al., 2019). We pursue this approach here and

[*]Equal contribution [1]Department of Industrial Engineering, Università degli Studi di Firenze, Firenze, Italy. Member of the INdAM Research Group GNCS. [2]Department of Mathematics and Computer Science "Ulisse Dini", Università degli Studi di Firenze, Firenze, Italy. Member of the INdAM Research Group GNCS. [3]Namur Center for Complex Systems (naXys), University of Namur, Namur, Belgium. Correspondence to: Gianmarco Gurioli <gianmarco.gurioli@unifi.it>, Stefania Bellavia <stefania.bellavia@unifi.it>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

our contributions are as follows. In (Bellavia et al., 2020b) adaptive regularisation methods with random models for computing strong approximate minimisers of any order for inexpensively constrained smooth optimization have been proposed. We focus here on the method from this class using second order models, and on the solution of unconstrained finite sum problems. While gradient and Hessian are subject to random noise ξ, function values are required to be approximated with deterministic level of accuracy. Our approach is then particularly suited for applications where evaluating derivatives is more expensive than performing function evaluations. This is for instance the case of deep neural networks training (see, e.g., (Goodfellow et al., 2016)). We discuss a matrix-free implementation in the case where gradient and Hessian are approximated via subsampling and with adaptive accuracy requirements. The outlined procedure retains the optimal worst-case complexity and our analysis complements that in (Cartis & Scheinberg, 2018), as we cover the approximation of second-order minimisers.

Notations. We use $\|\cdot\|$ to indicate the 2-norm (matrices and vectors). $\mathbb{E}[X]$ denotes the expected value of a random variable X. All inexact quantities are denoted by an overbar.

2. ARC with inexact evaluations

2.1. Preliminaries.

We make the following assumptions on (1).

AS.1 There exists a constant $f_{\rm low}$ such that $f(x) \geq f_{\rm low}$ for all $x \in \mathbb{R}^n$.

AS.2 $f \in \mathcal{C}^2(\mathcal{B})$ with $\mathcal{B}$ a convex neighbourhood of $\mathbb{R}^n$. Moreover, there exists a nonnegative constant L_H, such that, for all $x, y \in \mathcal{B}$: $\|\nabla_x^2 f(x) - \nabla_x^2 f(y)\| \leq L_H \|x - y\|$.

Consequently, the second order Taylor expansion of f centered at x with increment s is well-defined and given by $T_{f,2}(x,s) \stackrel{\rm def}{=} f(x) - \Delta T_{f,2}(x,s)$, where

$$\Delta T_{f,2}(x,s) \stackrel{\rm def}{=} -\nabla_x f(x)^\top s - \frac{1}{2} s^\top \nabla_x^2 f(x) s. \quad (2)$$

Similarly, $T_{f,1}(x,s) \stackrel{\rm def}{=} f(x) - \Delta T_{f,1}(x,s)$, with $\Delta T_{f,1}(x,s) \stackrel{\rm def}{=} -\nabla_x f(x)^\top s$.

2.2. Optimality conditions

To obtain complexity results for approximating second-order minimisers, we use a compact formulation to characterise the optimality conditions. As in (Cartis et al., 2020), given the order of accuracy $q \in \{1,2\}$, the tolerance vector $\epsilon = \epsilon_1$, if $q = 1$, or $\epsilon = (\epsilon_1, \epsilon_2)$, if $q = 2$, we say that $x \in \mathbb{R}^n$ is a q-th order ϵ-approximate minimiser for (1) if

$$\phi_{f,j}(x) \leq \frac{\epsilon_j}{j} \text{ for } j = 1, ..., q, \quad (3)$$

where

$$\phi_{f,j}(x) \stackrel{\rm def}{=} f(x) - \min_{\|d\| \leq 1} T_{f,j}(x,d) = \max_{\|d\| \leq 1} \Delta T_{f,j}(x,d), \quad (4)$$

The optimaliy measure $\phi_{f,j}$ is a nonnegative (continuous) function that can be used as a measure of closeness to q-th order stationary points (Cartis et al., 2020). For $\epsilon_1 = \epsilon_2 = 0$, it reduces to the known first- and second-order optimality conditions, respectively. Indeed, assuming that $q = 1$ we have from (3)–(4) with $j = 1$ that $\phi_{f,1}(x) = \|\nabla_x f(x)\| = 0$. If $q = 2$, (3)–(4) further imply that $\phi_{f,2}(x) = \max_{\|d\| \leq 1} \left(-\frac{1}{2} d^\top \nabla_x^2 f(x) d\right) = 0$, which is the same as requiring the semi-positive definiteness of $\nabla_x^2 f(x)$.

2.3. The Stochastic ARC ($SARC_q$) algorithm

We now define our Stochastic ARC scheme $SARC_q$, whose purpose is to find a q-th ϵ-approximate minimiser of (1) (see (3)). The scheme is defined in analogy with the basic ARC framework (see, e.g. (Cartis et al., 2011a)), but now uses the inexact values $\overline{f}(x_k)$, $\overline{f}(x_k + s)$, $\overline{\nabla_x^j f}(x_k)$ instead of $f(x_k)$, $f(x_k + s)$, $\nabla_x^j f(x_k)$, $j = 1, ..., q$, respectively. At iteration k, the computable regularised cubic model

$$m_k(s) \quad = -\overline{\Delta T}_{f,2}(x_k, s) + \tfrac{\sigma_k}{6}\|s\|^3 \quad (7)$$

is built and approximately minimised finding a step s_k such that

$$m_k(s_k) \leq m_k(0) = 0 \quad (8)$$

and

$$\overline{\phi}_{m_k,j}(s_k) = \max_{\|d\| \leq 1} \overline{\Delta T}_{m_k,j}(s_k, d) \leq \theta \frac{\epsilon_j}{j}, \quad (9)$$

for $j = 1, ..., q$ and some $\theta \in (0, \frac{1}{2})$, where

$$\begin{aligned} \overline{\Delta T}_{m_k,1}(s_k,d) &= -\overline{\nabla_x f}(x_k)^\top d - s_k^\top \overline{\nabla_x^2 f}(x_k) d \\ &\quad -\frac{\sigma_k}{2}\|s_k\| s_k^\top d \\ \overline{\Delta T}_{m_k,2}(s_k,d) &= \overline{\Delta T}_{m_k,1}(s_k,d) - \frac{1}{2} d^\top \overline{\nabla_x^2 f}(x_k) d \\ &\quad -\frac{\sigma_k}{2}\|s_k\|\|d\|^2. \end{aligned}$$

The existence of such a step can be proved as in Lemma 4.4 of (Cartis et al., 2020) (see also (Bellavia et al., 2020b)). We note that the model definition in (7) does not depend on the approximate value of f at x_k. The ratio ρ_k, depending on inexact function values and model, is then computed at Step 4 and affects the acceptance of the trial point $x_k + s_k$. Its magnitude also influences the regularisation parameter σ_k update at Step 5. While the gradient $\overline{\nabla_x f}(x_k)$ and the Hessian $\overline{\nabla_x^2 f}(x_k)$ approximations can be seen as random estimates, the values $\overline{f}(x_k)$, $\overline{f}(x_k + s_k)$ are required to be

Algorithm 1 The $SARC_q$ Algorithm

Step 0: Initialization. An initial point $x_0 \in \mathbb{R}^n$, initial regulariser $\sigma_0 > 0$, tolerances ϵ_j, $j = 1, ..., q$, and constants $\theta \in (0, \frac{1}{2})$, $\eta \in (0, 1)$, $\gamma > 1$, $\alpha \in (0, 1)$, $\omega_0 = \min\left[\frac{1}{2}\alpha\eta, \frac{1}{\sigma_0}\right]$, $\sigma_{\min} \in (0, \sigma_0)$ are given. Set $k = 0$.

Step 1: Model definition. Build the approximate gradient $\overline{\nabla_x f}(x_k)$ and Hessian $\overline{\nabla^2_x f}(x_k)$ and compute the model $m_k(s)$ as defined in (7).

Step 2: Step calculation. Compute a step s_k satisfying (8)–(9), for $j = 1, ..., q$. If $\overline{\Delta T}_{f,2}(x_k, s_k) = 0$, go to Step 4.

Step 3: Function approximations. Compute $\overline{f}(x_k)$ and $\overline{f}(x_k + s_k)$ satisfying (10)–(11).

Step 4: Test of acceptance. Set

$$\rho_k = \begin{cases} \frac{\overline{f}(x_k) - \overline{f}(x_k + s_k)}{\overline{\Delta T}_{f,2}(x_k, s_k)} & \text{if } \overline{\Delta T}_{f,2}(x_k, s_k) > 0, \\ -\infty & \text{otherwise.} \end{cases}$$

If $\rho_k \geq \eta$ (*successful iteration*), then set $x_{k+1} = x_k + s_k$; otherwise (*unsuccessful iteration*) set $x_{k+1} = x_k$.

Step 5: Regularisation parameter update. Set

$$\sigma_{k+1} = \begin{cases} \max\left[\sigma_{\min}, \frac{1}{\gamma}\sigma_k\right], & \text{if} \rho_k \geq \eta, \\ \gamma\sigma_k, & \text{if} \rho_k < \eta. \end{cases} \quad (5)$$

Step 6: Relative accuracy update. Set

$$\omega_{k+1} = \min\left[\frac{1}{2}\alpha\eta, \frac{1}{\sigma_{k+1}}\right]. \quad (6)$$

Increment k by one and go to Step 1.

deterministically computed to satisfy

$$\left|\overline{f}(x_k) - f(x_k)\right| \leq \omega_k \overline{\Delta T}_{f,2}(x_k, s_k), \quad (10)$$
$$\left|\overline{f}(x_k + s_k) - f(x_k + s_k)\right| \leq \omega_k \overline{\Delta T}_{f,2}(x_k, s_k), \quad (11)$$

in which ω_k is iteratively defined at Step 6. As for the implementation of the algorithm, we note that $\overline{\phi}_{m_k,1} = \|\nabla_s m_k(s_k)\|$, while $\overline{\phi}_{m_k,2}(s_k)$ can be computed via a standard trust-region method at a cost which is comparable to that of computing the Hessian left-most eigenvalue. The approximate minimisation of the cubic model (7) at each iteration can be seen as an issue in the ARC framework. However, an approximate minimiser can be computed via matrix-free approaches accessing the Hessian only through matrix-vector products. A number of procedures have been proposed, ranging from Lanczos-type iterations where the minimisation is done via nested, lower dimensional, Krylov subspaces (Cartis et al., 2011a), up to minimisation via gradient descent (see, e.g., (Agarwal et al., 2017; Carmon & Duchi, 2016; Carmon et al., 2018)) or the Barzilai-Borwein gradient method (Bianconcini et al., 2015). Hessian-vector products can be approximated by the finite difference approximation, with only two gradient evaluations (Bianconcini et al., 2015; Carmon et al., 2018). All these matrix-free implementations remain relevant if $\nabla^2 f(x_k)$ is defined via subsampling, proceeding as in Section 3.1 of (Berahas et al., 2020). Interestingly, back-propagation-like methods in deep learning also allow computations of Hessian-vector products at a similar cost (Pearlmutter, 1994; Schraudolph, 2002).

2.4. Probabilistic assumptions on $SARC_q$

In what follows, all random quantities are denoted by capital letters, while the use of small letters denotes their realisations. We refer to the random model M_k at iteration k, while $m_k = M_k(\zeta_k)$ is its realisation, with ζ_k being a random sample taken from a context-dependent probability space. As a consequence, the iterates X_k, as well as the regularisers Σ_k, the steps S_k and Ω_k, are the random variables such that $x_k = X_k(\zeta_k)$, $\sigma_k = \Sigma_k(\zeta_k)$, $s_k = S_k(\zeta_k)$ and $\omega_k = \Omega_k(\zeta_k)$. For the sake of brevity, we will omit ζ_k in what follows. Due to the randomness of the model construction at Step 1, the $SARC_q$ algorithm induces a random process formalised by $\{X_k, S_k, M_k, \Sigma_k, \Omega_k\}$. For $k \geq 0$, we formalise the conditioning on the past by using $\mathcal{A}^M_{k-1}$, the $\hat{\sigma}$-algebra induced by the random variables M_0, M_1,..., M_{k-1}, with $\mathcal{A}^M_{-1} = \hat{\sigma}(x_0)$. We also denote by $d_{k,j}$ and $\overline{d}_{k,j}$ the arguments in the maximum in the definitions of $\phi_{m_k,j}(s_k)$ and $\overline{\phi}_{m_k,j}(s_k)$, respectively. We say that iteration k is *accurate* if

$$\|\overline{\nabla^\ell_x f}(X_k) - \nabla^\ell_x f(X_k)\| \leq \Omega_k \frac{\overline{\Delta T}_{k,\min}}{6\tau^\ell_k}, \text{ for } \ell \in \{1, 2\}, \quad (12)$$

where

$$\tau_k \stackrel{\text{def}}{=} \max\left[\|S_k\|, \max_{j=1,...,q}[\|D_{k,j}\|, \|\overline{D}_{k,j}\|]\right]$$
$$\overline{\Delta T}_{k,\min} \stackrel{\text{def}}{=} \min\Big[\overline{\Delta T}_{f,p}(X_k, S_k), \min_{j=1,...,q}\Big[\overline{\Delta T}_{m_k,j}(S_k, D_{k,j}), \overline{\Delta T}_{m_k,j}(S_k, \overline{D}_{k,j})\Big]\Big].$$

We emphasize that the above accuracy requirements are adaptive. At variance with the trust-region methods of (Blanchet et al., 2019; Cartis & Scheinberg, 2018; Chen et al., 2018a), the above conditions do not need the model to be fully linear or quadratic in a ball centered at x_k of radius at least $\|s_k\|$. As standard in related works (Blanchet et al., 2019; Cartis & Scheinberg, 2018; Chen et al., 2018a; Paquette & Scheinberg, 2018), we assume a lower bound

on the probability of the model to be accurate at the k-th iteration.

AS.3 For all $k \geq 0$, conditioned to $\mathcal{A}_{k-1}^M$, we assume that (12) is satisfied with probability at least $p \in (\frac{1}{2}, 1]$ independent of k.

We stress that the inequalities in (12) can be satisfied via uniform subsampling with probability of success at least p_ℓ, $\ell \in \{1, 2\}$, using the operator Bernstein inequality (see, Section 7.2 in (Bellavia et al., 2019)), and this provides AS.3 with $p = p_1 p_2$.

3. Worst-case complexity analysis

3.1. Stopping time

Before tackling the worst-case evaluation complexity of the $SARC_q$ algorithm, it is important to point out the clarifications below. For each $k \geq 1$ we assume that the computation of s_{k-1} and thus of the trial point $x_{k-1} + s_{k-1}$ ($k \geq 1$) are deterministic, once the inexact model $m_{k-1}(s)$ is known, and that (10)-(11) at Step 3 of the algorithm are enforced deterministically; therefore, ρ_{k-1} and the fact that iteration $k-1$ is successful are deterministic outcomes of the realisation of the (random) inexact model. Hence,

$$N_\epsilon = \inf\Big\{k \geq 0 \ \mid \ \phi_{f,j}(X_k) \leq \frac{\epsilon_j}{j}, \ j = 1, ..., q\Big\}$$

can be seen as a family of hitting times depending of ϵ and corresponding to the number of iterations required until (3) is met for the first time. The assumptions made on top of this subsection imply that the variables $X_{k-1} + S_{k-1}$ and the event $\{X_k = X_{k-1} + S_{k-1}\}$, occurring when iteration $k-1$ is successful, are measurable with respect to $\mathcal{A}_{k-1}^M$. Our aim is then to derive an upper bound on the expected number of steps $\mathbb{E}[N_\epsilon]$ needed by the $SARC_q$ algorithm, in the worst-case, to reach an ϵ-approximate q-th-order-necessary minimiser, as in (3).

3.2. Deriving expected upper bounds on N_ϵ

A crucial property for our complexity analysis is to show that, when the model (7) is accurate, iteration k is successful but $\|\nabla_x f(x_{k+1})\| > \epsilon_1$, then $\|s_k\|^3$ is lower bounded by $\psi(\sigma_k)\epsilon_1^{3/2}$ in which $\psi(\sigma)$ is a decreasing function of σ. This is central in (Cartis & Scheinberg, 2018) for proving the worst-case complexity bound for first-order optimality. By virtue of the compact formulation (4) of the optimality measure, such a lower bound for $\|s_k\|^3$ also holds for second-order optimality and a suitable power of ϵ_2.

Lemma 1. Suppose that AS.2 holds and consider any realisation of the algorithm. Suppose that (12) also occurs, that iteration k is successful and that, for some $j = 1, ..., q$, (3) fails for x_{k+1}. Then $\|s_k\|^3 \geq \psi(\sigma_k)\epsilon_j^{\frac{3}{2}q}$, with

$$\psi(\sigma) = \min\left[1, \left(\frac{(1-2\theta)(3-q)}{q(L_{f,2}+\sigma)}\right)^{\frac{3}{2}q}\right]. \tag{13}$$

We refer to Lemma 3.4 in (Bellavia et al., 2020b) for a complete proof. In order to perform the complexity analysis, building on (Cartis & Scheinberg, 2018) and using results from the previous subsection, we first state a threshold σ_s for the regulariser, above which each iteration of the algorithm with accurate model is successful. We also give some preliminary bounds (see (Bellavia et al., 2020b)).

Lemma 2. Suppose that AS.2 holds. For any realisation of the algorithm, if iteration k is such that (12) occurs and

$$\sigma_k \geq \sigma_s \stackrel{\text{def}}{=} \max\left[\sigma_0, \frac{L_{f,2}+3}{1-\eta}\right], \tag{14}$$

then iteration k is successful.

Lemma 3 Let Assumptions AS.1–AS.3 hold and assume that $\Sigma_0 = \gamma^{-i}\sigma_s$ for some positive integer i. For all realisations of the $SARC_q$, let N_{AS}, N_Λ and N_{Λ^c} represent the number of accurate successful iterations with $\Sigma_k \leq \sigma_s$, the number of iterations with $\Sigma_k < \sigma_s$ and the number of iterations with $\Sigma_k \geq \sigma_s$, respectively. Then,

$$\begin{aligned}
\mathbb{E}[N_{\Lambda^c}] &\leq \frac{1}{2p}\mathbb{E}[N_\epsilon], \\
\mathbb{E}[N_{AS}] &\leq \frac{6(f_0 - f_{\text{low}})}{\eta\sigma_{\min}(1-\alpha)\psi(\sigma_s)} \max_{j=1,...,q}\left[\epsilon_j^{-\frac{3}{2}q}\right] + 1, \\
\mathbb{E}[N_\Lambda] &\leq \frac{1}{2p-1}\left[2\mathbb{E}[N_{AS}] + \log_\gamma\Big(\frac{\sigma_s}{\sigma_0}\Big)\right].
\end{aligned}$$

The proofs of the first and the third bound of the previous lemma follow the reasoning in (Cartis & Scheinberg, 2018), the proof of the second bound is given in (Bellavia et al., 2020b). Finally, the fact that $\mathbb{E}[N_\epsilon] = \mathbb{E}[N_\Lambda] + \mathbb{E}[N_{\Lambda^c}]$ allows us to state our complexity result.

Theorem 1. Under the assumptions of Lemma 3,

$$\begin{aligned}
\mathbb{E}[N_\epsilon] \leq \frac{2p}{(2p-1)^2}\Big[&\frac{12(f_0 - f_{\text{low}})}{\eta\sigma_{\min}(1-\alpha)\psi(\sigma_s)} \max_{j=1,...,q}\left[\epsilon_j^{-\frac{3}{2}q}\right] \\
&+ \log_\gamma\left(\frac{\sigma_s}{\sigma_0}\right) + 2\Big].
\end{aligned} \tag{15}$$

4. Conclusions and perspectives

The final expected bounds in (15) are sharp in the order of the tolerance ϵ. The effect of inaccurate evaluations is thus limited to scaling the optimal complexity we would otherwise derive from the deterministic analysis (see, e.g., Theorem 4.2 in (Bellavia et al., 2020a)), by a factor which depends on the probability p of the model being accurate.

The inclusion of inexact function evaluations subject to random noise is at the moment an open and challenging issue.

Acknowledgements

INdAM-GNCS partially supported the first and third authors under Progetti di Ricerca 2019 and 2020. The second author was partially supported by INdAM through a GNCS grant.

References

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1195–1199, 2017.

Bellavia, S., Gurioli, G., Morini, B., and Toint, P. L. Adaptive regularization algorithms with inexact evaluations for nonconvex optimization. *SIAM Journal on Optimization*, 29(4):2881–2915, 2019.

Bellavia, S., Gurioli, G., and Morini, B. Adaptive cubic regularization methods with dynamic inexact Hessian information and applications to finite-sum minimization. *IMA Journal of Numerical Analysis*, 2020a.

Bellavia, S., Gurioli, G., Morini, B., and Toint, P. L. High-order evaluation complexity of a stochastic adaptive regularization algorithm for nonconvex optimization using inexact function evaluations and randomly perturbed derivatives. *arXiv preprint arXiv:2005.04639*, 2020b.

Berahas, A. S., Bollapragada, R., and Nocedal, J. An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software*, pp. 1–20, 2020.

Bianconcini, T., Liuzzi, G., Morini, B., and Sciandrone, M. On the use of iterative methods in cubic regularization for unconstrained optimization. *Computational Optimization and Applications*, 60(1):35–57, 2015.

Birgin, E. G., Gardenghi, J., Martínez, J. M., Santos, S. A., and Toint, P. L. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163 (1-2):359–368, 2017.

Blanchet, J., Cartis, C., Menickelly, M., and Scheinberg, K. Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS journal on optimization*, 1(2):92–119, 2019.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Carmon, Y. and Duchi, J. C. Gradient descent efficiently finds the cubic-regularized non-convex Newton step. *arXiv preprint arXiv:1612.00547*, 2016.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

Cartis, C. and Scheinberg, K. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, 169 (2):337–375, 2018.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011a.

Cartis, C., Gould, N. I., and Toint, P. L. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130(2):295–319, 2011b.

Cartis, C., Gould, N. I., and Toint, P. L. Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108, 2012a.

Cartis, C., Gould, N. I., and Toint, P. L. An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity. *IMA Journal of Numerical Analysis*, 32(4):1662–1695, 2012b.

Cartis, C., Gould, N., and Toint, P. L. Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions. *arXiv preprint arXiv:2001.10802*, 2020.

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

Chen, X., Jiang, B., Lin, T., and Zhang, S. On adaptive cubic regularized Newton's methods for convex optimization via random sampling. *arXiv preprint arXiv:1802.05426*, 2018b.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Kohler, J. M. and Lucchi, A. Sub-sampled cubic regularization for non-convex optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1895–1904. JMLR. org, 2017.

Paquette, C. and Scheinberg, K. A stochastic line search method with convergence rate analysis. *arXiv preprint arXiv:1807.07994*, 2018.

Pearlmutter, B. A. Fast exact multiplication by the Hessian. *Neural computation*, 6(1):147–160, 1994.

Schraudolph, N. N. Fast curvature matrix-vector products for second-order gradient descent. *Neural computation*, 14(7):1723–1738, 2002.

Xu, P., Roosta, F., and Mahoney, M. W. Newton-type methods for non-convex optimization under inexact Hessian information. *Mathematical Programming*, pp. 1–36, 2019.

Xu, P., Roosta, F., and Mahoney, M. W. Second-order optimization for non-convex machine learning: An empirical study. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pp. 199–207. SIAM, 2020.

Yao, Z., Xu, P., Roosta-Khorasani, F., and Mahoney, M. W. Inexact non-convex Newton-type methods. *arXiv preprint arXiv:1802.06925*, 2018.

Zhou, D., Xu, P., and Gu, Q. Stochastic variance-reduced cubic regularization methods. *Journal of Machine Learning Research*, 20(134):1–47, 2019.