Random-subspace adaptive cubic regularisation method for nonconvex optimisation

Zhen Shao SHAOZ@MATHS.OX.AC.UK
Coralia Cartis CARTIS@MATHS.OX.AC.UK
Mathematical Institute, University of Oxford.

Abstract

We investigate second-order methods for nonconvex optimisation, and propose a Random Subspace Adaptive Cubic Regularisation (R-ARC) method, which we analyse under various assumptions on the objective function and the sketching matrices that generate the random subspaces. We show that, when the sketching matrix achieves a subspace embedding of the augmented matrix of the gradient and the Hessian with sufficiently high probability, then the R-ARC method satisfies, with high probability, a complexity bound of order $\mathcal{O}\left(\epsilon^{-3/2}\right)$ to drive the (full) gradient norm below ϵ; matching in the accuracy order its deterministic counterpart (ARC). As an illustration, we particularise our results to the special case of a scaled Gaussian ensemble.

1. Introduction

High-order methods for smooth nonconvex optimization problems use second- or higher-order derivative information at each iteration to determine the next iterate, in addition to gradient information. Using problem information beyond first-order typically results in faster rates of convergence of the ensuing methods, and attainment of second-order (or higher) optimality conditions [2, 6], allowing the avoidance of (strict) saddle points in machine learning applications. However, the use of higher order derivative information increases the computational costs per iteration, especially when derivative evaluations are expensive. Compared with first-order methods which need at most d partial/componentwise derivatives, higher-order methods need to evaluate d^2 or even d^3 partial derivatives at each iteration, where d is the number of variables in the objective. For many machine learning models, especially deep neural networks, d is commonly in the millions, making higher order optimisation methods inapplicable.

Subspace methods aim to alleviate this high computational cost by using only randomly sampled/projected derivative information at each iteration, thus reducing the dimensionality of the parameter space to a manageable magnitude. For example, the (basic) coordinate descent method [10] uses only one partial derivative at each iteration, instead of the full gradient. More generally, subspace methods make use of the theoretical guarantees offered by Johnson-Lindenstrauss (JL) Lemma and related random embedding constructions, where the full gradient is replaced by a randomly embedded gradient in lower dimensions with approximately the same norm. A generic first-order random subspace framework based on random embeddings is analysed in [3, 8].

The purpose of this paper is to extend the analysis based on random embeddings in [3, 8] to adaptive cubically regularised second-order methods, for which we propose a subspace variant (R-ARC). We show almost-sure global convergence of R-ARC to a first-order stationary point for non-convex objectives, with a convergence rate matching the full space counterpart, which is the optimal rate for second order methods [4]. Moreover, a variety of random embeddings, including Gaussian, countsketch or subsampling can be applied to the gradient and the second derivatives of the objective function.

Although the analysis here makes use of the generic framework in [3, 8], in this work we make extensive use of *subspace* embedding properties of random ensembles, not just the pointwise JL-type needed in [3, 8]. The former are needed due to a particular aspect of the analysis of cubic regularization methods, namely, the presence, at the current iteration, of the gradient evaluated at the next (random) iterate. Controlling this random quantity has proved challenging and we are not aware of other existing results in a sketching framework and for nonconvex objectives. Note that the randomized cubic regularization frameworks in [5, 11–13] are designed for sums of functions and sketching in the observational domain, while the cubic regularization variant with probabilistic models in [1] is more restrictive compared to our framework and anaylsis (see (9) and related comments).

2. R-ARC: random subspace adaptive cubic regularisation method

We first describe the random subspace adaptive cubic regularization algorithm (R-ARC). The algorithm builds on the deterministic ARC algorithm [2, 6], by replacing the full gradient and the full Hessian with their projections onto a random subspace. At each iteration, a random matrix S_k is drawn (whose desired properties are to be discussed), with which a random subspace model is built around the current iterate. The model is approximately minimised in the subspace and the minimiser is projected back into the full space, in order to obtain a full-dimensional trial step s_k, which is then either accepted or rejected, by evaluating the function decrease/increase.

Algorithm 1 Random subspace cubic regularisation algorithm (R-ARC)

Initialization Choose a matrix distribution $\mathcal{S}$ of matrices $S \in \mathbb{R}^{l \times d}$. Choose constants $\gamma_1 \in (0,1)$, $\gamma_2 > 1$, $\theta \in (0,1)$, $\kappa_T \geq 0$ and $\alpha_{\max} > 0$ such that $\gamma_2 = \frac{1}{\gamma_1^c}$, for some $c \in \mathbb{N}^+$. Initialize the algorithm by setting $x_0 \in \mathbb{R}^d$, $\alpha_0 = \alpha_{max}\gamma_1^p$ for some $p \in \mathbb{N}^+$ and $k = 0$.

1. Compute a reduced model and a trial step

Draw a random matrix $S_k \in \mathbb{R}^{l \times d}$ from $\mathcal{S}$, and let

$$\begin{aligned} \hat{m}_k(\hat{s}) &= f(x_k) + \langle S_k \nabla f(x_k), \hat{s} \rangle + \frac{1}{2}\langle \hat{s}, S_k \nabla^2 f(x_k) S_k^T \hat{s} \rangle + \frac{1}{3\alpha_k} \left\| S_k^T \hat{s} \right\|_2^3 \\ &= \hat{q}_k(\hat{s}) + \frac{1}{3\alpha_k} \left\| S_k^T \hat{s} \right\|_2^3, \end{aligned} \tag{1}$$

where $\hat{q}_k(\hat{s})$ is the second order Taylor series of $f(x_k + S_k^T \hat{s}_k)$ around x_k. Compute $\hat{s}_k$ by approximately minimising (1) such that

$$\hat{m}_k(\hat{s}_k) \leq \hat{m}_k(0) \quad \text{and} \quad \|\nabla \hat{m}_k(\hat{s}_k)\|_2 \leq \kappa_T \left\| S_k^T \hat{s}_k \right\|_2^2. \tag{2}$$

Compute a trial step

$$s_k = w_k(\hat{s}_k) = S_k^T \hat{s}_k, \tag{3}$$

2. Check sufficient decrease

Check sufficient decrease as defined by the condition

$$f(x_k) - f(x_k + s_k) \geq \theta \left[\hat{q}_k(0) - \hat{q}_k(\hat{s})\right], \tag{4}$$

3. Update the parameter α_k and possibly take the trial step s_k

If (4) holds, set $x_{k+1} = x_k + s_k$ and $\alpha_{k+1} = \min\{\alpha_{max}, \gamma_2 \alpha_k\}$ [successful iteration].
Otherwise, set $x_{k+1} = x_k$ and $\alpha_{k+1} = \gamma_1 \alpha_k$ [unsuccessful iteration].
In either case, let $k = k + 1$.

Remark 1 *Two main strategies for computing $\hat{s}_k$ by minimising* (1) *are given in [2], either requiring a factorisation of $S_k \boldsymbol{\nabla}^2 f(x_k) S_k^T$ (in a Newton-like algorithm), or repeated matrix-vector products involving $S_k \boldsymbol{\nabla}^2 f(x_k) S_k^T$ (in a Lanczos-based algorithm). Therefore, in addition to requiring only projected derivatives, R-ARC also significantly reduces the computation of the trial step by reducing the dimension of the linear systems involved in its solution from $d \times d$ in the full dimensional case, to $l \times l$ in the subspace case.*

3. Worst-case complexity of the R-ARC algorithm

The main result of this paper shows that for suitably generated matrices S_k, R-ARC produces an iterate x_k with $\|\nabla f(x_k)\|_2 \leq \epsilon$ in $N = \mathcal{O}\left(\epsilon^{-3/2}\right)$ iterations, with high probability. The *true* iterations of R-ARC are iterations such that S_k successfully captures key properties of the iterates[1].

Definition 2 *Let $\epsilon_S \in (0,1)$, $S_{max} > 0$. Iteration k is (ϵ_S, S_{max})-true if*

$$\|S_k M_k z_k\|_2^2 \geq (1 - \epsilon_S) \|M_k z_k\|_2^2, \quad \textit{for all } z_k \in \mathbb{R}^{d+1} \tag{5}$$

$$\|S_k\|_2 \leq S_{max}, \tag{6}$$

where[2] $M_k = \begin{bmatrix} \boldsymbol{\nabla} f(x_k) & \boldsymbol{\nabla}^2 f(x_k) \end{bmatrix} \in \mathbb{R}^{d \times (d+1)}$.

Theorem 3 *Let $\delta_S \in (0,1), \epsilon_S \in (0,1), S_{max} > 0$ with $\delta_S < \frac{c}{(c+1)^2}$, where c is defined in Algorithm 1. Suppose that S_k satisfies the following condition: for any $\bar{x}_k \in \mathbb{R}^d$, $k = 1, 2, \ldots$, we have $\mathbb{P}\left(T_k | x_k = \bar{x}_k\right) \geq 1 - \delta_S$, where $\mathbb{P}\left(T_0\right) \geq 1 - \delta_S$ and*

$$T_k = \begin{cases} 1, & \textit{if iteration } k \textit{ is } (\epsilon_S, S_{max})\textit{-true} \\ 0, & \textit{otherwise}. \end{cases} \tag{7}$$

Let $\delta_1 \in (0,1)$ such that $g(\delta_S, \delta_1) = \left[(1 - \delta_S)(1 - \delta_1) - 1 + \frac{c}{(c+1)^2}\right]^{-1} > 0$. Suppose that f is twice continuously differentiable, with an L_H-Lipschitz continuous Hessian. Run Algorithm 1 for minimizing f for N iterations. Then, for any $\epsilon \in (0,1)$, if $N = \mathcal{O}\left(\epsilon^{-3/2} \max\left(L_H^{3/2}, S_{\max}^{3/2}\right)\right)$, we have

$$\mathbb{P}\left(\min_{k \leq N} \|\boldsymbol{\nabla} f(x_k)\|_2 \leq \epsilon\right) \geq 1 - e^{-\frac{\delta_1^2 (1 - \delta_S) N}{2}}.$$

[1] There is an interplay between the success probability of S_k and the constant c in Algorithm 1, as we shall see.

[2] Note that all vectors are column vectors.

A few remarks are in order. Firstly, we note that our complexity result holds with high probability, namely, it provides (lim inf) global convergence almost surely. Secondly, the convergence result only guarantees at least one past iterate satisfying $\|\nabla f(x_k)\|_2 \leq \epsilon$. However the non-monotonicity of the gradient is alleviated because R-ARC produces monotonically decreasing $f(x_k)$, due to Step 2, and so the function value $f(x_N)$ is at least as small as a function value with a small gradient. Finally, the *true* assumptions on S_k are satisfied if S_k is drawn from a distribution known as an oblivious subspace embedding, defined below.

Definition 4 (ϵ_S-subspace embedding [9]) *An ϵ_S-subspace embedding for (the column subspace of) a matrix $H \in \mathbb{R}^{d\times k}$ is a matrix $S \in \mathbb{R}^{l\times d}$ such that*

$$(1-\epsilon)\|y\|_2^2 \leq \|Sy\|_2^2 \leq (1+\epsilon)\|y\|_2^2, \quad \textit{for all } y \in Y = \{y = Hz : z \in \mathbb{R}^k\}. \tag{8}$$

Oblivious subspace embeddings are matrix distributions such that given a(ny) column subspace of vectors in $\mathbb{R}^n$, a random matrix drawn from such a distribution is an embedding for these vectors with high probability. We let $1-\tilde{\delta} \in [0,1]$ denote a(ny) success probability of an embedding.

Definition 5 (Oblivious subspace embedding [7, 9]) *A distribution $\mathcal{S}$ on $S \in \mathbb{R}^{m\times n}$ is an $(\epsilon_S, \tilde{\delta})$-oblivious subspace embedding if, given a fixed/arbitrary matrix $H \in \mathbb{R}^{d\times k}$, a matrix S from the distribution is an ϵ_S-subspace embedding for H, with probability at least $1-\tilde{\delta}$.*

Using the above definitions of embeddings, we have that if S_k are drawn from an oblivious subspace embedding distribution for the matrices $M \in \mathbb{R}^{d\times(d+1)}$ with rank at most $r+1$, where r is the maximum rank of the Hessian of the iterates $\nabla^2 f(x_k)$, then (5) is satisfied with high conditional probability at each iteration[3]. The other condition for true iterations, namely, (6), is satisfied (with high probability) by a variety of random matrices. We give an example of distributions from which S_k may be drawn that satisfy the assumptions in Theorem 3. We note that many other matrice ensembles are oblivious subspace embeddings and probabilistically bounded above, see [8].

4. R-ARC with scaled Gaussian sketching matrices

A choice for S_k is the scaled Gaussian matrix, defined below.

Definition 6 *We say $S \in \mathbb{R}^{l\times d}$ is a scaled Gaussian matrix if S_{ij} are independently distributed as $N(0, l^{-1})$.*

The following two properties of Gaussian matrices are well known.

Lemma 7 (Theorem 2.3 in [9]) *Let $\epsilon_S \in (0,1)$ and $S \in \mathbb{R}^{l\times d}$ be a scaled Gaussian matrix with $l = \mathcal{O}\left(\epsilon_S^{-2} r \log\left(\frac{1}{\delta_S^{(1)}}\right)\right)$ Then the distribution of S is an $(\epsilon_S, \delta_S^{(1)})$-oblivious subspace embedding for $d\times(d+1)$ matrices M with ranks at most $r+1$.*

Lemma 8 *Let $S \in \mathbb{R}^{l\times d}$ be a scaled Gaussian matrix. Then with probability $1-\delta_S^{(2)}$,*

$$\|S\|_2 \leq 1 + \sqrt{\frac{d}{l}} + \sqrt{\frac{2\log\left(1/\delta_S^{(2)}\right)}{l}}.$$

[3] However S_k being drawn from an oblivious subspace embedding is not a necessary condition for the assumptions in Theorem 3 to hold.

Thus, if S_k is a scaled Gaussian matrix, the conditions in Theorem 3 hold with $\delta_S = \delta_S^{(1)} + \delta_S^{(2)}$, by taking a union bound, with S_{max} given by in the last lemma. We note that l, the dimension of the subspace, does not depend on d but is proportional to r, the rank of the Hessian. Thus, for any meaningful dimensionality reduction to take place, it is essential that the analysis of R-ARC is applied to an objective f with low-rank Hessians. However, in [8], an alternative condition on the Hessian is derived, namely, the Hessian is highly sparse except for a few row entries (that is, the function f only varies significantly over a few directions).

Comparison with related work We achieve the same $\mathcal{O}\left(\epsilon^{-3/2}\right)$ convergence rate as in [1], which is optimal for non-convex optimization using second order models [4], and matches the deterministic ARC method. A key difference between our work and [1] is the definition of true iterations. Instead of Theorem 2, [1] define true iterations as those iterations that satisfy

$$\|\boldsymbol{\nabla} f(x_k) - \boldsymbol{\nabla} m_k(s_k)\|_2 \leq \kappa_g \|s_k\|_2^2 \tag{9}$$

for some constant $\kappa_g > 0$.

This difference leads to potentially-distinct applications of the two frameworks. To construct the model m_k, [1] proposed to use sampling with adaptive sample sizes for problems having the finite sum structure ($f = \sum_i f_i$), or to use finite differences in the context of derivative-free optimisation. However, without other assumptions, even just in order to obtain condition (9), one may need a sample size that may be impractically large. By contrast, in our framework, the sketching size is fixed, of order 1, and even then, true iterations occur sufficiently frequently for scaled Gaussian matrices (and indeed for other random embeddings).

Inexact local models constructed by subsampling for sums of functions have also been proposed for cubic regularization and other Newton-type methods in [5, 11–13]. Our emphasis here is related to reducing specifically the dimension of the variable domain (rather than the observational space).

5. Conclusion

In this paper we present a random subspace variant (R-ARC) of the adaptive cubic regularization algorithm for nonconvex problems. We show that under embedding assumptions for the random subspace, R-ARC achieves the same worst case convergence rate as the full-space variant. We note that in [8], we also showed R-ARC converges to a second order critical point with the same worst case convergence rate as the full-space variant.

6. Acknowledgement

This work was supported by the EPSRC Centre For Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with the Numerical Algorithms Group Ltd; and by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA). Zhen Shao's work was also supported by the EPSRC Doctoral Prize (EP/T517811/1(D4T00230)).

References

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

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

[3] C. Cartis, J. Fiala, and Z. Shao. Randomised subspace methods for non-convex optimization, with applications to nonlinear least-squares. *arXiv e-prints, in preparation*, 2022.

[4] C. Cartis, N. I. M. Gould, and P. L. Toint. *Evaluation complexity of algorithms for nonconvex optimization*. MOS-SIAM series on Optimization. Society for Industrial and Applied Mathematics (SIAM), 2022.

[5] J. M. Kohler and A. Lucchi. Sub-sampled cubic regularization for non-convex optimization. *arXiv e-prints*, art. arXiv:705.05933, May 2017. URL `https://arxiv.org/abs/705.05933`.

[6] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Math. Program.*, 108(1, Ser. A):177–205, 2006. ISSN 0025-5610. doi: 10.1007/s10107-006-0706-8. URL `https://doi.org/10.1007/s10107-006-0706-8`.

[7] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152, 2006. doi: 10.1109/FOCS.2006.37.

[8] Z. Shao. On random embeddings and their application to optimisation. *arXiv e-prints*, art. arXiv:2206.03371, June 2022.

[9] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. ISSN 1551-305X. doi: 10.1561/0400000060. URL `https://doi.org/10.1561/0400000060`.

[10] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151:3—34, 2015.

[11] P. Xu, F. Roosta-Khorasan, and M. W. Mahoney. Newton-type methods for non-convex optimization under inexact Hessian information. *arXiv e-prints*, art. arXiv:1708.07164, Aug 2017. URL `https://arxiv.org/abs/1708.07164`.

[12] P. Xu, F. Roosta-Khorasan, and M. W. Mahoney. Second-order optimization for non-convex machine learning: An empirical study. *arXiv e-prints*, art. arXiv:1708.07827, Aug 2017. URL `https://arxiv.org/abs/1708.07827`.

[13] Z. Yao, P. Xu, F. Roosta-Khorasan, and M. W. Mahoney. Inexact non-convex Newton-type methods. *arXiv e-prints*, art. arXiv:1802.06925, Feb 2018. URL `https://arxiv.org/abs/1802.06925`.