Sub-sampled Cubic Regularization for Non-convex Optimization

Jonas Moritz Kohler [1] **Aurelien Lucchi** [1]

Abstract

We consider the minimization of non-convex functions that typically arise in machine learning. Specifically, we focus our attention on a variant of trust region methods known as cubic regularization. This approach is particularly attractive because it escapes *strict* saddle points and it provides stronger convergence guarantees than first- and second-order as well as classical trust region methods. However, it suffers from a high computational complexity that makes it impractical for large-scale learning. Here, we propose a novel method that uses sub-sampling to lower this computational cost. By the use of concentration inequalities we provide a sampling scheme that gives sufficiently accurate gradient and Hessian approximations to retain the strong global and local convergence guarantees of cubically regularized methods. To the best of our knowledge this is the first work that gives global convergence guarantees for a sub-sampled variant of cubic regularization on non-convex functions. Furthermore, we provide experimental results supporting our theory.

1. Introduction

In this paper we address the problem of minimizing an objective function of the form

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right], \qquad (1)$$

where $f(\mathbf{x}) \in C^2(\mathbb{R}^d, \mathbb{R})$ is a not necessarily convex, (regularized) loss function over n datapoints. Stochastic Gradient Descent (SGD) is a popular method to optimize this type of objective especially in the context of large-scale learning when n is very large. Its convergence properties are well understood for convex functions, which arise in many machine learning applications (Nesterov, 2004). However, non-convex functions are also ubiquitous and have recently drawn a lot of interest due to the growing success of deep neural networks. Yet, non-convex functions are extremely hard to optimize due to the presence of saddle points and local minima which are not global optima (Dauphin et al., 2014; Choromanska et al., 2015). In fact, the work of (Hillar & Lim, 2013) showed that even a degree four polynomial can be NP-hard to optimize. Instead of aiming for a global minimizer, we will thus seek for a local optimum of the objective. In this regard, a lot of attention has focused on a specific type of functions known as strict saddle functions or ridable functions (Ge et al., 2015; Sun et al., 2015). These functions are characterized by the fact that the Hessian of every saddle point has a negative eigenvalue. Geometrically this means that there is a direction of negative curvature where decreasing function values can be found. Examples of strict saddle functions include dictionary learning, orthogonal tensor decomposition and generalized phase retrieval (Ge et al., 2015; Sun et al., 2015).

In this work, we focus our attention on trust region methods to optimize Eq. 1. These methods construct and optimize a local model of the objective function within a region whose radius depends on how well the model approximates the real objective. One of the keys for efficiency of these methods is to pick a model that is comparably easy to optimize, such as a quadratic function (Conn et al., 2000). Following the trust region paradigm, cubic regularization methods (Nesterov & Polyak, 2006; Cartis et al., 2011a) suggest finding the step $\mathbf{s}_k$ that minimizes a cubic model of the form

$$m_k(\mathbf{s}_k) := f(\mathbf{x}_k) + \mathbf{s}_k^\intercal \nabla f(\mathbf{x}_k) + \frac{1}{2}\mathbf{s}_k^\intercal \mathbf{H}_k \mathbf{s}_k + \frac{\sigma_k}{3}\left\|\mathbf{s}_k\right\|^3, \qquad (2)$$

where $\mathbf{H}_k := \nabla^2 f(\mathbf{x}_k)$ and $\sigma_k > 0$ [1].

(Nesterov & Polyak, 2006) were able to show that, if the

[1] Department of Computer Science, ETH Zurich, Switzerland. Correspondence to: Jonas Moritz Kohler <jonas.kohler@student.kit.edu>, Aurelien Lucchi <aurelien.lucchi@inf.ethz.ch>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1] In the work of (Nesterov & Polyak, 2006), σ_k is assumed to be the Lipschitz constant of the Hessian in which case the model defined in Eq. 2 is a global overestimator of the objective, i.e. $f(\mathbf{x}) \leq m(\mathbf{x})\ \forall \mathbf{x} \in \mathbb{R}^d$. We will elaborate on the role of σ_k in (Cartis et al., 2011a) later on.

step is computed by globally minimizing the cubic model and if the Hessian $\mathbf{H}_k$ is globally Lipschitz continuous, Cubic regularization methods possess the best known worst case complexity to solve Eq. 1: an overall worst-case iteration count of order $\epsilon^{-3/2}$ for generating $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$, and of order ϵ^{-3} for achieving approximate nonnegative curvature. However, minimizing Eq. 2 in an exact manner impedes the performance of this method for machine learning applications as it requires access to the full Hessian matrix. More recently, (Cartis et al., 2011a) presented a method (hereafter referred to as ARC) which relaxed this requirement by assuming that one can construct an approximate Hessian $\mathbf{B}_k$ that is sufficiently close to $\mathbf{H}_k$ in the following way:

$$\|(\mathbf{B}_k - \mathbf{H}_k)\mathbf{s}_k\| \leq C\,\|\mathbf{s}_k\|^2,\ \forall k \geq 0, C > 0 \quad (3)$$

Furthermore, they showed that it is sufficient to find an approximate minimizer by applying a Lanczos method to build up evolving Krylov spaces, which can be constructed in a Hessian-free manner, i.e. by accessing the Hessians only indirectly via matrix-vector products. However there are still two obstacles for the application of ARC in the field of machine learning: (1) The cost of the Lanczos process increases linearly in n and is thus not suitable for large datasets and (2) there is no theoretical guarantee that quasi-Newton approaches satisfy Eq. 3 and (Cartis et al., 2011a) do not provide any alternative approximation technique.

In this work, we make explicit use of the finite-sum structure of Eq. 1 by applying a sub-sampling technique in order to provide guarantees for machine learning applications. Towards this goal, we make the following contributions:

- We provide a theoretical Hessian sampling scheme that is guaranteed to satisfy Eq. 3 with high probability.
- We extend the analysis to inexact gradients and prove that the convergence guarantees of (Nesterov & Polyak, 2006; Cartis et al., 2011a) can be retained.
- Since the dominant iteration cost lie in the construction of the Lanczos process and increase linearly in n, we lower the computational cost significantly by reducing the number of samples used in each iteration.
- Finally, we provide experimental results demonstrating significant speed-ups compared to standard first and second-order optimization methods for various convex and non-convex objectives.

2. Related work

Sampling techniques for first-order methods. In large-scale learning, when $n \gg d$ most of the computational cost of traditional deterministic optimization methods is spent in computing the exact gradient information. A common technique to address this issue is to use sub-sampling in order to compute an unbiased estimate of the gradient. The simplest instance is Stochastic Gradient Descent (SGD) whose convergence does not depend on the number of datapoints n. However, the variance in the stochastic gradient estimates slows its convergence down. The work of (Friedlander & Schmidt, 2012) explored a sub-sampling technique for gradient descent in the case of convex functions, showing that it is possible to maintain the same convergence rate as full-gradient descent by carefully increasing the sample size over time. Another way to recover a linear rate of convergence for strongly-convex functions is to use variance-reduced methods (Johnson & Zhang, 2013; Defazio et al., 2014; Roux et al., 2012; Hofmann et al., 2015; Daneshmand et al., 2016). Recently, the convergence of SGD and its variance-reduced counterparts has also been extended to non-convex functions (Ghadimi & Lan, 2013; Reddi et al., 2016a) but the techniques used in these papers require using a randomized sampling scheme which is different from what is typically used in practice. Furthermore, the guarantees these methods provide are only in terms of convergence to critical points. However, the work of (Ge et al., 2015; Sun et al., 2015) recently showed that SGD can achieve stronger guarantees in the case of strict saddle functions. Yet, the convergence rate has a polynomial dependency to the dimension d and the smallest eigenvalue of the Hessian which can make this method fairly impractical.

Second-order methods. For second-order methods, the problem of avoiding saddle points is even worse as they might be attracted by saddle points or even points of local maximizers (Dauphin et al., 2014). Another predominant issue is the computation (and perhaps storage) of the Hessian matrix, which requires $O(nd^2)$ operations as well as computing the inverse of the Hessian, which requires $O(d^3)$ computations. Quasi-Newton methods such as the well-known (L-)BFGS algorithm partially address this issue by requiring $O(nd + d^2)$ per-iteration cost (Nesterov, 2004) instead of $O(nd^2 + d^3)$. An increasingly popular alternative is to use sub-sampling techniques to approximate the Hessian matrix, such as done for example in (Byrd et al., 2011) and (Erdogdu & Montanari, 2015). The latter method, named NewSamp, approximates the Hessian with a low-rank approximation which reduces the complexity per iteration to $O(nd + |S|d^2)$ with $|S|$ being the sample size [2]. Although this is a significant reduction in terms of complexity, NewSamp yields a composite convergence rate: quadratic at first but only linear near the minimizer. Unlike NewSamp, our sampling scheme yields a locally quadratic rate of convergence (as well as faster global con-

[2] Note that this method still requires $O(nd)$ computation for the gradient as it only subsamples the Hessian.

vergence). Our analysis also does not require using exact gradients and can thus further reduce the complexity per iteration.

Cubic regularization and trust region methods. Trust region methods are among the most effective algorithmic frameworks to avoid pitfalls such as local saddle points in non-convex optimization. Classical versions iteratively construct a local quadratic model and minimize it within a certain radius wherein the model is trusted to be sufficiently similar to the actual objective function. This is equivalent to minimizing the model function with a suitable *quadratic* penalty term on the stepsize. Thus, a natural extension is the cubic regularization method introduced by (Nesterov & Polyak, 2006) that uses a *cubic* over-estimator of the objective function as a regularization technique for the computation of a step to minimize the objective function. The drawback of their method is that it requires computing the exact minimizer of Eq. 2, thus requiring the exact gradient and Hessian matrix. However finding a global minimizer of the cubic model $m_k(\mathbf{s})$ may not be essential in practice and doing so might be prohibitively expensive from a computational point of view. (Cartis et al., 2011a) introduced a method named ARC which relaxed this requirement by letting $\mathbf{s}_k$ be an approximation to the minimizer. The model defined by the adaptive cubic regularization method introduced two further changes. First, instead of computing the exact Hessian $\mathbf{H}_k$ it allows for a symmetric approximation $\mathbf{B}_k$. Second, it introduces a dynamic positive parameter σ_k instead of using the global Lipschitz constant L.

There have been efforts to further reduce the computational complexity of this problem. For example, (Agarwal et al., 2016) refined the approach of (Nesterov & Polyak, 2006) to return an approximate local minimum in time which is linear in the input representation. Similar improvements have been made by (Carmon & Duchi, 2016) and (Hazan & Koren, 2016). These methods provide alternatives to minimize the cubic model and can thus be seen as complementary to our approach. Finally, the work of (Blanchet et al., 2016) proposed a stochastic variant of a trust region method but their analysis focused on randomized gradients only.

3. Formulation

We are interested in optimizing Eq. 1 in a large-scale setting when the number of datapoints n is very large such that the cost of solving Eq. 2 exactly becomes prohibitive. In this regard we identify a sampling scheme that allows us to retain the convergence results of deterministic trust region and cubic regularization methods, including quadratic local convergence rates and global second-order convergence guarantees as well as worst-case complexity bounds. A detailed theoretical analysis is given in Section 4. Here we shall first state the algorithm itself and elaborate further on the type of local nonlinear models we employ as well as how these can be solved efficiently.

3.1. Objective function

Instead of using deterministic gradient and Hessian information as in Eq. 2, we use unbiased estimates of the gradient and Hessian constructed from two independent sets of points denoted by S_g and S_B. We then construct a local cubic model that is (approximately) minimized in each iteration:

$$m_k(\mathbf{s}_k) := f(\mathbf{x}_k) + \mathbf{s}_k^\intercal \mathbf{g}_k + \frac{1}{2}\mathbf{s}_k^\intercal \mathbf{B}_k \mathbf{s}_k + \frac{\sigma_k}{3}\|\mathbf{s}_k\|^3 \quad (4)$$

where $\mathbf{g}_k := \frac{1}{|S_g|}\sum_{i\in S_g} \nabla f_i(\mathbf{x}_k)$

and $\quad \mathbf{B}_k := \frac{1}{|S_B|}\sum_{i\in S_B} \nabla^2 f_i(\mathbf{x}_k)$.

The model derivative with respect to $\mathbf{s}_k$ is defined as:

$$\nabla m_k(\mathbf{s}_k) = \mathbf{g}_k + \mathbf{B}_k\mathbf{s}_k + \lambda \mathbf{s}_k \text{ ,where } \lambda = \sigma_k \|\mathbf{s}_k\|. \quad (5)$$

3.2. Algorithm

Our Sub-sampled Cubic Regularization approach (SCR) is presented in Algorithm 1. At iteration step k, we subsample two sets of datapoints from which we compute a stochastic estimate of the gradient and the Hessian. We then solve the problem in Eq. 4 *approximately* using the method described in Section 3.4 and update the regularization parameter σ_k depending on how well the model approximates the real objective. In particular, very successful steps indicate that the model is (at least locally) an adequate approximation of the objective such that the penalty parameter is decreased in order to allow for longer steps. For unsuccessful iterations we proceed exactly the opposite way. Readers familiar with trust region methods might see that one can interpret the penalty parameter σ_k as inversely proportional to the trust region radius δ_k.

3.3. Exact model minimization

Solving Eq. 4 requires minimizing an unconstrained non-convex problem that may have isolated local minima. As shown in (Cartis et al., 2011a) the global model minimizer $\mathbf{s}_k^*$ is characterized by following systems of equations,

$$(\mathbf{B}_k + \lambda_k^*\mathbf{I})\mathbf{s}_k^* = -\mathbf{g}_k,\ \lambda_k^* = \sigma_k\|\mathbf{s}_k^*\|, (\mathbf{B}_k + \lambda_k^*\mathbf{I}) \succeq 0. \quad (9)$$

In order to find a solution we can express $\mathbf{s}_k^* := \mathbf{s}_k(\lambda_k^*) = -(\mathbf{B}_k + \lambda_k^*\mathbf{I})^{-1}\mathbf{g}_k$, apply this in the second equation of (9) and obtain a univariate, nonlinear equation in λ_k

$$\left\|-(\mathbf{B}_k + \lambda_k^*\mathbf{I})^{-1}\mathbf{g}_k\right\| - \frac{\lambda_k^*}{\sigma_k} = 0. \quad (10)$$

Algorithm 1 Sub-sampled Cubic Regularization (SCR)

```
Input:
    Starting point $\mathbf{x}_0 \in \mathbb{R}^d$ (e.g $\mathbf{x}_0 = \mathbf{0}$)
    $\gamma > 1, 1 > \eta_2 > \eta_1 > 0$, and $\sigma_0 > 0$
for $k = 0, 1, \ldots,$ until convergence do
    Sample gradient $\mathbf{g}_k$ and Hessian $\mathbf{H}_k$ according to Eq. 17 & Eq. 19 respectively
    Obtain $\mathbf{s}_k$ by solving $m_k(\mathbf{s}_k)$ (Eq. 4) such that A1 holds
    Compute $f(\mathbf{x}_k + \mathbf{s}_k)$ and
        $\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{f(\mathbf{x}_k) - m_k(\mathbf{s}_k)}$    (6)
    Set
        $\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k \geq \eta_1 \\ \mathbf{x}_k & \text{otherwise} \end{cases}$    (7)
    Set
        $\sigma_{k+1} = \begin{cases} \max\{\min\{\sigma_k, \|\mathbf{g}_k\|\}, \epsilon_m\} & \text{if } \rho_k > \eta_2 \text{ (very successful iteration)} \\ \sigma_k & \text{if } \eta_2 \geq \rho_k \geq \eta_1 \text{ (successful iteration)} \\ \gamma\sigma_k & \text{otherwise (unsuccessful iteration),} \end{cases}$    (8)
    where $\epsilon_m \approx 10^{-16}$ is the relative machine precision.
end for
```

Furthermore, we need $\lambda_k^* \geq \max\{-\lambda_1(\mathbf{B}_k), 0\}$, where $\lambda_1(\mathbf{B}_k)$ is the leftmost eigenvalue of $\mathbf{B}_k$, in order to guarantee the semi-positive definiteness of $(\mathbf{B}_k + \lambda_k^*\mathbf{I})$.

Thus, computing the global solution of m_k boils down to finding the root of Eq. 10 in the above specified range of λ_k. The problem can be solved by Newton's method, which involves factorizing $\mathbf{B}_k + \lambda_k\mathbf{I}$ for various λ_k and is thus prohibitively expensive for large problem dimensions d. See Section 6.2 in (Cartis et al., 2011a) for more details. In the following Section we instead explore an approach to *approximately* minimize the model while retaining the convergence guarantees of the exact minimization.

3.4. Approximate model minimization

(Cartis et al., 2011a) showed that it is possible to retain the remarkable properties of the cubic regularization algorithm with an inexact model minimizer. A necessary condition is that $\mathbf{s}_k$ satisfies the two requirements stated in A1.

Assumption 1 (Approximate model minimizer)**.**

$$\mathbf{s}_k^\intercal\mathbf{g}_k + \mathbf{s}_k^\intercal\mathbf{B}_k\mathbf{s}_k + \sigma_k \|\mathbf{s}_k\|^3 = 0 \tag{11}$$

$$\mathbf{s}_k^\intercal\mathbf{B}_k\mathbf{s}_k + \sigma_k \|\mathbf{s}_k\|^3 \geq 0 \tag{12}$$

Note that the first equation is equal to $\nabla_s m_k(\mathbf{s}_k)^\intercal\mathbf{s}_k = 0$ and the second to $\mathbf{s}_k^\intercal\nabla_s^2 m_k(\mathbf{s}_k)\mathbf{s}_k \geq 0$.

As shown in (Cartis et al., 2011a) Lemma 3.2, the global minimizer of $m_k(\mathbf{s}_k)$ in a Krylov subspace $\mathcal{K}_k := \text{span}\{\mathbf{g}_k, \mathbf{H}_k\mathbf{g}_k, \mathbf{H}_k^2\mathbf{g}_k, ...\}$ satisfies this assumption independent of the subspace dimension. This comes in handy, as minimizing m_k in the Krylov subspace only involves factorizing a tri-diagonal matrix, which can be done at the cost of $O(d)$. However, a Lanczos-type method must be used in order to build up an orthogonal basis of this subspace which typically involves one matrix-vector product ($O((2d-1)n)$) per additional subspace dimension (see Chapter 5 in (Conn et al., 2000) for more details).

Thus, in order to keep the per iteration cost of SCR low and in accordance to ARC, we apply the following termination criterion to the Lanczos process in the hope to find a suitable trial step before $\mathcal{K}_k$ is of dimensionality d.

Assumption 2 (Termination Criteria)**.** *For each outer iteration k, assume that the Lanczos process stops as soon as some Lanczos iteration i satisfies the criterion*

$$\textit{TC:}\ \|\nabla m_k(\mathbf{s}_{i,k})\| \leq \theta_k \|\mathbf{g}_k\|, \tag{13}$$

where $\theta_k = \kappa_\theta \min(1, \|\mathbf{s}_{i,k}\|)$, $\kappa_\theta \in (0, 1)$.

However, we argue that especially for high dimensional problems, the cost of the Lanczos process may significantly slow down cubically regularized methods and since this cost increases linearly in n, carefully sub-sampled versions are an attractive alternative.

4. Theoretical analysis

In this section, we provide the convergence analysis of SCR. For the sake of brevity, we assume Lipschitz continuous Hessians right away but note that a superlinear local convergence result as well as the global first-order convergence theorem can both be obtained without the former assumption.

First, we lay out some critical assumptions regarding the gradient and Hessian approximations. Second, we show that one can theoretically satisfy these assumptions with high probability by sub-sampling first- and second-order information. Third, we give a condensed convergence analysis of SCR which is widely based on (Cartis et al., 2011a), but adapted for the case of stochastic gradients. There, we show that the local and global convergence properties of ARC can be retained by sub-sampled versions at the price of slightly worse constants.

4.1. Assumptions

Assumption 3 (Continuity)**.** *The functions $f_i \in C^2(\mathbb{R}^d)$, g_i and H_i are Lipschitz continuous for all i, with Lipschitz constants κ_f, κ_g and κ_H respectively.*

By use of the triangle inequality, it follows that these assumptions hold for all $\mathbf{g}$ and $\mathbf{H}$, independent of the sample size. Furthermore, note that the Hessian and gradient norms are uniformly bounded as a consequence of A3.

In each iteration, the Hessian approximation $\mathbf{B}_k$ shall satisfy condition AM.4 from (Cartis et al., 2011a), which we restate here for the sake of completeness.

Assumption 4 (Sufficient Agreement of $\mathbf{H}$ and $\mathbf{B}$)**.**

$$\|(\mathbf{B}_k - \mathbf{H}(\mathbf{x}_k))\mathbf{s}_k\| \leq C \left\|\mathbf{s}_k\right\|^2, \forall k \geq 0, C > 0. \quad (14)$$

We explicitly stress the fact that this condition is stronger than the well-known Dennis Moré Condition. While quasi-Newton approximations satisfy the latter, there is no theoretical guarantee that they also satisfy the former (Cartis et al., 2011a). Furthermore, any sub-sampled gradient shall satisfy the following condition.

Assumption 5 (Sufficient Agreement of ∇f and g)**.**

$$\|\nabla f(\mathbf{x}_k) - \mathbf{g}(\mathbf{x}_k)\| \leq M \left\|\mathbf{s}_k\right\|^2, \forall k \geq 0, \ M > 0. \quad (15)$$

4.2. Sampling Conditions

Based on probabilistic deviation bounds for random vectors and matrices[3], we now present sampling conditions that guarantee sufficient steepness and curvature information in each iteration k. In particular, the Bernstein inequality gives exponentially decaying bounds on the probability of a random variable to differ by more than ϵ from its mean for any fixed number of samples. We use this inequality to upper bound the ℓ_2-norm distance $\|\nabla f - \mathbf{g}\|$, as well as the spectral-norm distance $\|\mathbf{B} - \mathbf{H}\|$ by quantities involving the sample size $|S|$. By applying the resulting bounds in the sufficient agreement assumptions (A4 & A5) and rearranging for $|S|$, we are able to translate the latter into concrete sampling conditions.

4.2.1. Gradient Sampling

As detailed in the Appendix, the following Lemma arises from the Vector Bernstein Inequality.

Lemma 6 (Gradient deviation bound)**.** *Let the sub-sampled gradient $\mathbf{g}_k$ be defined as in Eq. 4. For $\epsilon \leq 2\kappa_f$ we have with probability $(1-\delta)$ that*

$$\|\mathbf{g}(\mathbf{x}_k) - \nabla f(\mathbf{x}_k)\| \leq 4\sqrt{2}\kappa_f \sqrt{\frac{\log((2d)/\delta) + 1/4}{|S_{g,k}|}}. \quad (16)$$

It constitutes a non-asymptotic bound on the deviation of the gradient norms that holds with high probability. Note how the accuracy of the gradients increases in the sample size. This bound yields the following condition.

Theorem 7 (Gradient Sampling)**.** *If*

$$|S_{g,k}| \geq \frac{32\kappa_f^2 \left(\log\left((2d)/\delta\right) + 1/4\right)}{M^2 \left\|\mathbf{s}_k\right\|^4}, \ M \geq 0, \forall k \geq 0 \quad (17)$$

then $\mathbf{g}_k$ satisfies the sufficient agreement condition A5 with probability $(1-\delta)$.

4.2.2. Hessian Sampling

In analogy to the gradient case, we use the matrix version of Bernstein's Inequality to derive the following Lemma.

Lemma 8 (Hessian deviation bound)**.** *Let the sub-sampled Hessian $\mathbf{B}$ be defined as in Eq. 4. For $\epsilon \leq 4\kappa_g$ we have with probability $(1-\delta)$ that*

$$\|\mathbf{B}(\mathbf{x}_k) - \mathbf{H}(\mathbf{x}_k)\| \leq 4\kappa_g \sqrt{\frac{\log(2d/\delta)}{|S_{B,k}|}}, \quad (18)$$

This, in turn, can be used to derive a Hessian sampling condition that is guaranteed to satisfy the sufficient agreement condition (A4) with high probability.

Theorem 9 (Hessian Sampling)**.** *If*

$$|S_{B,k}| \geq \frac{16\kappa_g^2 \log(2d/\delta)}{(C\left\|\mathbf{s}_k\right\|)^2}, \ C \geq 0, \textit{ and } \forall k \geq 0 \quad (19)$$

[3] These bounds have lately become popular under the name of *concentration inequalities*. Unlike classic limit theorems, such as the Central Limit Theorem, concentration inequalities are specifically attractive for application in machine learning because of their non-asymptotic nature.

then $\mathbf{B}_k$ *satisfies the strong agreement condition A4 with probability* $(1-\delta)$.

As expected, the required sample sizes grow in the problem dimensionality d and in the Lipschitz constants κ_f and κ_g. Finally, as outlined in the Appendix (Lemma 20), the stepsize tends to zero as SCR approaches a second-order critical point. Consequently, the sample sizes must approach n as the algorithm converges and thus we have

$$\mathbf{g} \to \nabla f \text{ as well as } \mathbf{B} \to \mathbf{H} \text{ as } k \to \infty \tag{20}$$

4.3. Convergence Analysis

The entire analysis of cubically regularized methods is prohibitively lengthy and we shall thus establish only the crucial properties that ensure global, as well as fast local convergence and improve the worst-case complexity of these methods over standard trust region approaches. Next to the cubic regularization term itself, these properties arise mainly from the penalty parameter updates and step acceptance criteria of the ARC framework, which give rise to a good relation between regularization and stepsize. Further details can be found in (Cartis et al., 2011a).

4.3.1. Preliminary Results

First, we note that the penalty parameter sequence $\{\sigma_k\}$ is guaranteed to stay within some bounded positive range, which is essentially due to the fact that SCR is guaranteed to find a successful step as soon as the penalty parameter exceeds some critical value σ_{sup}.

Lemma 10 (Boundedness of σ_k). *Let A3, A4 and A5 hold. Then*

$$\sigma_k \in [\sigma_{\text{inf}}, \sigma_{\text{sup}}],\ \forall k \geq 0, \tag{21}$$

where σ_{inf} *is defined in Step 7 of Algorithm 1 and*

$$\sigma_{sup} := \{\sigma_0, \frac{3}{2}\gamma_2(2M + C + \kappa_g)\}. \tag{22}$$

Furthermore, for any successful iteration the objective decrease can be directly linked to the model decrease via the step acceptance criterion in Eq. 8. The latter, in turn, can be shown to be lower bounded by the stepsize which combined gives the following result.

Lemma 11 (Sufficient function decrease). *Suppose that* $\mathbf{s}_k$ *satisfies A1. Then, for all successful iterations* $k \geq 0$

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \eta_1(f(\mathbf{x}_k) - m(\mathbf{s}_k)) \\ &\geq \frac{1}{6}\eta_1\sigma_{\text{inf}}\,\|\mathbf{s}_k\|^3 \end{aligned} \tag{23}$$

Finally, the termination criterion (13) also guarantees step sizes that do not become too small compared to the respective gradient norm which leads to the following Lemma.

Lemma 12 (Sufficiently long steps). *Let A3, A4 and A5 hold. Furthermore, assume the termination criterion TC (A2) and suppose that* $\mathbf{x}_k \to \mathbf{x}^*$*, as* $k \to \infty$*. Then, for all sufficiently large successful iterations,* $\mathbf{s}_k$ *satisfies*

$$\|\mathbf{s}_k\| \geq \kappa_s \sqrt{\|\nabla f(\mathbf{x}_{k+1})\|} \tag{24}$$

where κ_s *is the positive constant*

$$\kappa_s = \sqrt{\frac{1-\kappa_\theta}{\frac{1}{2}\kappa_g + (1+\kappa_\theta\kappa_g)M + C + \sigma_{\text{sup}} + \kappa_\theta\kappa_g}}. \tag{25}$$

4.3.2. Local convergence

We here provide a proof of local convergence for any sampling scheme that satisfies the conditions presented in Theorem 7 and Theorem 9 as well as the additional condition that the sample size does not decrease in unsuccessful iterations. We show that such sampling schemes eventually yield exact gradient and Hessian information. Based upon this observation, we obtain the following local convergence result (as derived in the Appendix).

Theorem 13 (Quadratic local convergence). *Let A3 hold and assume that* $\mathbf{g}_k$ *and* $\mathbf{B}_k$ *are sampled such that 17 and 19 hold and* $|S_{g,k}|$ *and* $|S_{B,k}|$ *are not decreased in unsuccessful iterations. Furthermore, let* s_k *satisfy A1 and*

$$\mathbf{x}_k \to \mathbf{x}^*,\ as\ k \to \infty, \tag{26}$$

where $\mathbf{H}(\mathbf{x}^*)$ *is positive definite. Moreover, assume the stopping criterion TC (A2). Then,*

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq c,\ c > 0 \ as\ k \to \infty\ (w.h.p.). \tag{27}$$

That is, $\mathbf{x}_k$ *converges in q-quadratically to* $\mathbf{x}^*$ *as* $k \to \infty$ *with high probability.*

4.3.3. Global convergence to first-order critical point

Lemma 10 and 11 allow us to lower bound the function decrease of a successful step in terms of the *full* gradient ∇f_k (as we will shorty detail in Eq. 31). Combined with Lemma 10, this allows us to give deterministic global convergence guarantees using only stochastic first order information.

Theorem 14 (Convergence to 1st-order Critical Points). *Let A1, A3, A4 and A5 hold. Furthermore, let $\{f(\mathbf{x}_k)\}$ be bounded below by some $f_{\inf} > -\infty$. Then*

$$\lim_{k\to\infty} \|\nabla f(\mathbf{x}_k)\| = 0 \tag{28}$$

4.3.4. Global convergence to second-order critical point

Unsurprisingly, the second-order convergence guarantee relies mainly on the use of second-order information so that the stochastic gradients do neither alter the result nor the proof as it can be found in Section 5 of (Cartis et al., 2011a). We shall restate it here for the sake of completeness.

Theorem 15 (Second-order global convergence). *Let A3, A4 and A5 hold. Furthermore, let $\{f(\mathbf{x}_k)\}$ be bounded below by $f_{\inf}$ and $\mathbf{s}_k$ be a global minimizer of m_k over a subspace $\mathcal{L}_k$ that is spanned by the columns of the $d \times l$ orthogonal matrix $\mathbf{Q}_k$. As $\mathbf{B} \to \mathbf{H}$ asymptotically (Eq. 20), any subsequence of negative leftmost eigenvalues $\{\lambda_{\min}(\mathbf{Q}_k^\intercal \mathbf{H}(\mathbf{x}_k)\mathbf{Q}_k)\}$ converges to zero for sufficiently large, successful iterations. Hence*

$$\lim_{k\in\mathcal{S}} \inf_{k\to\infty} \lambda_{\min}(\mathbf{Q}_k^\intercal \mathbf{H}(\mathbf{x}_k)\mathbf{Q}_k) \geq 0. \tag{29}$$

Finally, if $\mathbf{Q}_k$ becomes a full orthogonal basis of $\mathbb{R}^d$ as $k \to \infty$, then any limit point of the sequence of successful iterates $\{\mathbf{x}_k\}$ is second-order critical (provided such a limit point exists).

4.3.5. Worst-case iteration complexity

For the worst-case analysis we shall establish the two disjoint index sets $\mathcal{U}_j$ and $\mathcal{S}_j$, which represent the un- and successful SCR iterations that have occurred up to some iteration $j > 0$, respectively. As stated in Lemma 10 the penalty parameter σ_k is bounded above and hence SCR may only take a limited number of consecutive unsuccessful steps. As a consequence, the total number of unsuccessful iterations is at most a problem dependent constant times the number of successful iterations.

Lemma 16 (Number of unsuccessful iterations). *For any fixed $j \geq 0$, let Lemma 10 hold. Then we have that*

$$|\mathcal{U}_j| \leq \left\lceil (|\mathcal{S}_j| + 1) \frac{\log(\sigma_{\sup}) - \log(\sigma_{\inf})}{\log(\eta_1)} \right\rceil. \tag{30}$$

Regarding the number of successful iterations we have already established the two key ingredients: (i) a sufficient function decrease in each successful iteration (Lemma 11) and (ii) a step size that does not become too small compared to the respective gradient norm (Lemma 12), which is essential to driving the latter below ϵ at a fast rate. Combined they give rise to the guaranteed function decrease for successful iterations

$$f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \geq \frac{1}{6}\eta_1 \sigma_{\inf} \kappa_s^3 \, \|\nabla f(\mathbf{x}_{k+1})\|^{3/2}, \tag{31}$$

which already contains the power of 3/2 that appears in the complexity bound. Finally, by summing over all successful iterations one obtains the following, so far best know, worst-case iteration bound to reach ϵ first-order criticality.

Theorem 17 (First-order worst-case complexity). *Let A1, A3, A4 and A5 hold. Furthermore, be $\{f(\mathbf{x}_k)\}$ bounded below by $f_{\inf}$ and TC applied (A2). Then, for $\epsilon > 0$ the total number of iterations* SCR *takes to generate the first iterate j with $\|\nabla f(\mathbf{x}_{j+1})\| \leq \epsilon$, and assuming $\epsilon \leq 1$, is*

$$j \leq \left\lceil (1+\kappa_i)(2+\kappa_j)\epsilon^{-3/2} \right\rceil, \tag{32}$$

where

$$\kappa_i = 6\frac{f(\mathbf{x}_0) - f_{\inf}}{\eta_1 \sigma_{\inf} \kappa_s^3} \text{ and } \kappa_j = \frac{\log(\sigma_{\sup}) - \log(\sigma_{\inf})}{\log(\eta_1)} \tag{33}$$

Note that the constants κ_i and κ_j involved in this upper bound both increase in the gradient inaccuracy M and the Hessian inaccuracy C (via κ_s and $\sigma_{\sup}$), such that more inaccuracy in the sub-sampled quantities may well lead to an increased overall number of iterations.

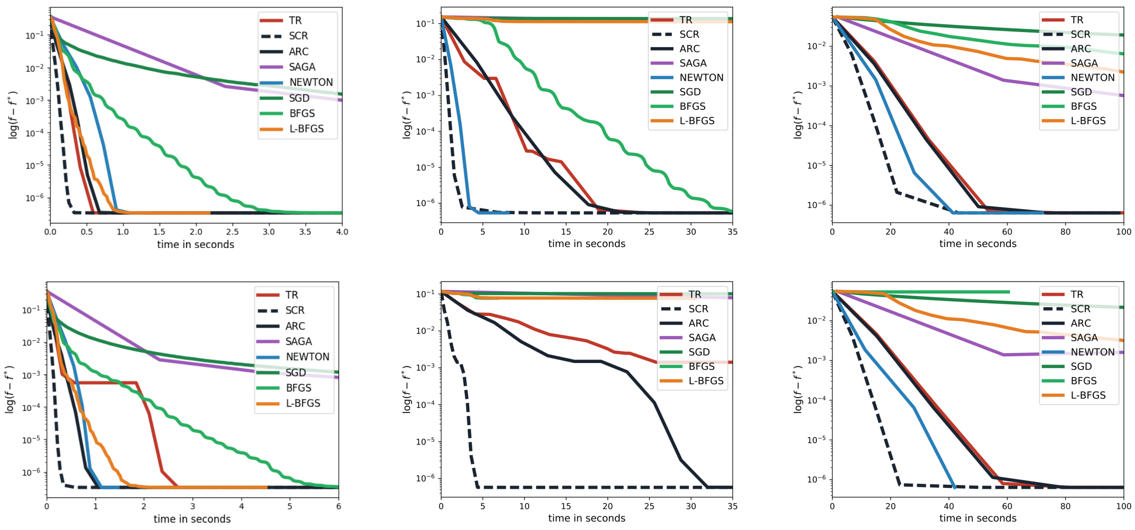
Finally, we want to point out that similar results can be established regarding a second-order worst-case complexity bound similar to Corollary 5.5 in (Cartis et al., 2011b), which we do not prove here for the sake of brevity.

5. Experimental results

In this section we present experimental results on real-world datasets where $n \gg d \gg 1$. They largely confirm the analysis derived in the previous section. Please refer to the Appendix for more detailed results and experiments on higher dimensional problems.

5.1. Practical implementation of SCR

We implement SCR as stated in Algorithm 1 and note the following details. Following (Erdogdu & Montanari, 2015), we require the sampling conditions derived in Section 4 to hold with probability $O(1 - 1/d)$, which yields



1. A9A (n=32561, d=123) 2. COVTYPE (n=581012, d=54) 3. HIGGS (n=11000000, d=28)

Figure 1. Top (bottom) row shows log suboptimality of convex (non-convex) regularized logistic regressions over time (avg. of 10 runs).

the following practically applicable sampling schemes

$$
\begin{aligned}
|S_{k,H}| &\geq \frac{36\kappa_g^2 \log(d)}{(C\,\|\mathbf{s}_k\|)^2},\ C > 0,\ \forall k > 0 \\
|S_{k,g}| &\geq \frac{32\kappa_f^2(\log(d) + 1/4)}{M^2\|\mathbf{s}_k\|^4},\ M > 0,\ \forall k > 0.
\end{aligned}
\tag{34}
$$

The positive constants C and M can be used to scale the sample size to a reasonable portion of the entire dataset and can furthermore be used to offset κ_g and κ_f, which are generally expensive to obtain.

However, when choosing $|S|$ for the current iteration k, the stepsize $\mathbf{s}_k$ is yet to be determined. Based on the Lipschitz continuity of the involved functions, we argue that the previous stepsize is a fair estimator of the current one and this is confirmed by experimental results. Finally, we would like to point out that the sampling schemes derived in Eq. 34 gives our method a clear edge over sampling schemes that do not take any iteration information into account, e.g. linearly or geometrically increased samples.

5.2. Baselines and datasets

We compare SCR to various optimization methods presented in Section 2. This includes SGD (with constant step-size), SAGA, Newton's method, BFGS, L-BFGS and ARC. More details concerning the choice of the hyper-parameters are provided in the appendix. We ran experiments on the datasets *a9a*, *covtype* and *higgs* (see details in the appendix). We experimented with a binary logistic regression model with two different regularizers: a standard ℓ_2 penalty $\lambda\|x\|^2$, and a non-convex regularizer $\lambda\sum_{i=1}^{d} x_{(i)}^2 / \left(1 + x_{(i)}^2\right)$ (see (Reddi et al., 2016b)).

5.3. Results

The results in Figure 1 confirm our intuition that SCR can reduce ARCs computation time without losing its global convergence property. Newton's method is the closest in terms of performance. However, it suffer heavily from an increase in d as can be seen by additional results provided in the appendix. Furthermore, it cannot optimize the non-convex version of *covtype* due to a singular Hessian. Notably, BFGS terminates early on the non-convex *higgs* dataset due to a local saddle point. Finally, the high condition number of *covtype* has a significant effect on the performance of SGD, SAGA and L-BFGS.

6. Conclusion

In this paper we proposed a sub-sampling technique to estimate the gradient and Hessian in order to construct a cubic model analogue to trust region methods. We show that this method exhibits the same convergence properties as its deterministic counterpart, which are the best known worst-case convergence properties on non-convex functions. Our proposed method is especially interesting in the large scale regime when $n \gg d$. Numerical experiments on both real and synthetic datasets demonstrate the performance of the proposed algorithm which we compared to its deterministic variant as well as more classical optimization methods. As future work we would like to explore the adequacy of our method to train neural networks which are known to be hard to optimize due to the presence of saddle points.

References

Agarwal, Naman, Allen-Zhu, Zeyuan, Bullins, Brian, Hazan, Elad, and Ma, Tengyu. Finding local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*, 2016.

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

Byrd, Richard H, Chin, Gillian M, Neveitt, Will, and Nocedal, Jorge. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

Carmon, Yair and Duchi, John C. Gradient descent efficiently finds the cubic-regularized non-convex newton step. *https://arxiv.org/abs/1612.00547*, 2016.

Cartis, Coralia, Gould, Nicholas IM, and Toint, Philippe L. Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011a.

Cartis, Coralia, Gould, Nicholas IM, and Toint, Philippe L. Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. *Mathematical programming*, 130 (2):295–319, 2011b.

Chang, Chih-Chung and Lin, Chih-Jen. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Choromanska, Anna, Henaff, Mikael, Mathieu, Michael, Arous, Gérard Ben, and LeCun, Yann. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Conn, Andrew R, Gould, Nicholas IM, and Toint, Philippe L. *Trust region methods*. SIAM, 2000.

Daneshmand, Hadi, Lucchi, Aurélien, and Hofmann, Thomas. Starting small - learning with adaptive sample sizes. In *International Conference on Machine Learning*, 2016.

Dauphin, Yann N, Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, Ganguli, Surya, and Bengio, Yoshua. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pp. 2933–2941, 2014.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Erdogdu, Murat A and Montanari, Andrea. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems*, pp. 3052–3060, 2015.

Friedlander, Michael P and Schmidt, Mark. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

Ge, Rong, Huang, Furong, Jin, Chi, and Yuan, Yang. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *COLT*, pp. 797–842, 2015.

Ghadimi, Saeed and Lan, Guanghui. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Gould, Nicholas IM, Porcelli, M, and Toint, Philippe L. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Computational optimization and applications*, 53(1):1–22, 2012.

Gross, David. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

Hazan, Elad and Koren, Tomer. A linear-time algorithm for trust region problems. *Mathematical Programming*, 158 (1-2):363–381, 2016.

Hillar, Christopher J and Lim, Lek-Heng. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60 (6):45, 2013.

Hofmann, Thomas, Lucchi, Aurelien, Lacoste-Julien, Simon, and McWilliams, Brian. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems 28*, pp. 2296–2304. Curran Associates, Inc., 2015.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

Nesterov, Yurii. Introductory lectures on convex optimization. applied optimization, vol. 87, 2004.

Nesterov, Yurii and Polyak, Boris T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Reddi, Sashank J, Hefny, Ahmed, Sra, Suvrit, Poczos, Barnabas, and Smola, Alex. Stochastic variance reduction for nonconvex optimization. *arXiv preprint arXiv:1603.06160*, 2016a.

Reddi, Sashank J, Sra, Suvrit, Póczos, Barnabás, and Smola, Alex. Fast incremental method for nonconvex optimization. *arXiv preprint arXiv:1603.06159*, 2016b.

Roux, Nicolas L, Schmidt, Mark, and Bach, Francis R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2012.

Sun, Ju, Qu, Qing, and Wright, John. When are nonconvex problems not scary? *arXiv preprint arXiv:1510.06096*, 2015.