Sub-Sampled Cubic Regularization for Non-Convex Optimization

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Sub-Sampled Cubic Regularization for Non-Convex Optimization

Master Thesis

Jonas Moritz Kohler

Monday 10th April, 2017

Advisors: Prof. Dr. T. Hofmann, Dr. A. Lucchi
Department of Computer Science, ETH Zürich

Abstract

This thesis explores the analysis and design of a new stochastic, iterative second-order method for minimizing unconstrained and continuous optimization problems as they typically arise in the field of large scale learning. Specifically, we consider the minimization of non-convex objectives and focus our attention on second-order trust region approaches that are commonly used to globalize Newton's method. The starting point of our analysis is a recent trust region variant known as cubic regularization, which is particularly attractive because it is guaranteed to escape (strict) saddle points and yields fast local and stronger global convergence rates than first and second-order linear-search, as well as classical trust-region methods. However, this method suffers from a high computational complexity which makes it impractical for large-scale learning. Here, we propose a novel approach that uses sub-sampling to lower this computational cost. By the use of non-asymptotic probabilistic deviation bounds we provide sampling schemes that give sufficiently accurate gradient and Hessian approximations to retain the remarkable global and local convergence guarantees of deterministic, cubically regularized methods. To the best of our knowledge this is the first work that gives global second-order guarantees as well as quadratic local convergence for a sub-sampled newton method and it is also the first to study Hessian sampling in the context of cubic regularization frameworks. Finally, we provide experimental results on well-known machine learning datasets that widely confirm our theoretical analysis.

Contents

Chapter 1

Introduction

The thesis at hand explores the analysis and design of a new stochastic, iterative second-order method for minimizing unconstrained and continuous optimization problems as they typically arise in the field of large scale learning. Specifically, we consider the minimization of (potentially non-convex, twice continuously differentiable) objective functions $F : \mathbb{R}^d \to \mathbb{R}$ which can be written as a finite sum of individual functions f_i that each correspond to a certain datapoint $i \in [0, 1, \ldots, n]$. Thus, we consider minimizing the following problem

$$\min_{w \in \mathbb{R}^d} F(w) := \frac{1}{n} \sum_{i=1}^{n} f_i(w). \tag{1.1}$$

in a setting where $n \gg d \gg 0$. Classical examples of such problems are logistic regressions, support vectors machine and neural networks training, tensor decomposition and many more. Optimization algorithms typically applied for solving (1.1) are iterative methods essentially walking down an error surface in a step wise manner. Thus, when considering the computational complexity of these methods, two things come into play: the total number of iterations an algorithm takes and the per-iteration cost of each step. Classical optimization methods tend to have high per-iteration cost and thus progress slowly on large scale problems.

However, with continuously increasing amounts of readily-available data in today's digital world, the need to solve optimization problems of unprecedented sizes has grown strongly. This development has triggered a lot of research giving rise to modern variants and modifications of these methods that scale better to instances with higher numbers of datapoints. The most famous example is likely the evolution from Gradient Descent over Stochastic Gradient Descent[1] to newer, variance-reduced versions like SVRG or SAGA

[1] which has per-iteration cost that are independent of n but suffers from slower convergence

that retain Gradient Descent's linear (asymptotic) convergence rate while keeping the iteration cost low.

We here want to contribute to a similar development that recently emerged in the field of second-order methods, where it has become increasingly popular to use sub-sampling techniques to approximate the Hessian matrix, such as done for example in [Byrd et al., 2011] and [Martens, 2010]. The latter was the first to apply a sub-sampled Newton method for deep learning and showed promising empirical advantages over training deep neural networks with first-order methods. Yet, their analysis is lacking theoretical convergence guarantees. The method we propose for minimizing (1.1) is based on the cubic regularization framework of Nesterov and Polyak [2006] as well as on a recently popularized adaptive version (ARC) that was introduced by Cartis et al. [2011]. These methods can be viewed as an extension of the well-known Trust Region approaches (e.g. [Conn et al., 2000]). They are particularly attractive because they escapes *strict* saddle points[2] and provide stronger convergence guarantees than first and second-order linear-search, as well as classical trust-region methods.

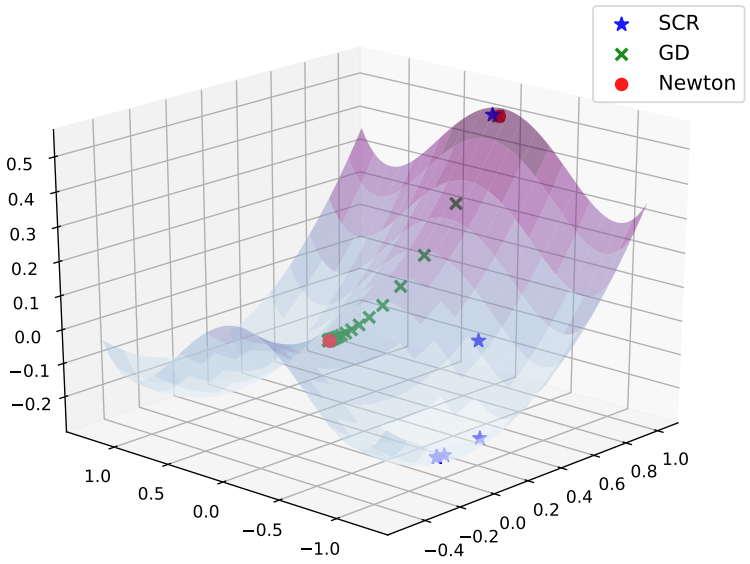


Figure 1.1: Graph of a two-dimensional, non-convex objective and path of the iterates of Gradient Descent (GD), Newton's method (Newton) as well as the stochastic cubic regularization method (SCR)

However, like most methods that incorporate second-order information they suffer from a high computational complexity which makes them impractical

[2]We will explain this term in Section 2.1.

for large-scale learning. Here, we propose a novel method that uses subsampling to lower this computational cost. Particularly, by the use of concentration inequalities, we derive iteration specific sampling conditions that yield sufficiently accurate gradient and Hessian approximations to retain the remarkable global and local convergence guarantees of deterministic cubic regularization methods.

The contributions of this thesis is fourfold:

- We provide gradient and Hessian sampling schemes that give rise to a stochastic cubic regularization method and prove that the convergence guarantees of [Nesterov and Polyak, 2006, Cartis et al., 2011] can be retained.
- At the same time we lower the computational cost significantly by reducing the number of samples used in each iteration.
- We provide experimental results demonstrating significant speed-ups compared to standard first and second-order optimization methods for various convex and non-convex objectives.
- Finally, to the best of our knowledge, this is the first work to apply Hessian sampling in a Trust Region framework and the first stochastic Newton method to achieve quadratic local convergence and a global second-order worst case complexity of $O(\varepsilon^{-3/2})$.

The thesis is structured as follows. To lay the foundation of the following analysis we firstly introduce the notation and give a short overview of the basic concepts and methods involved in unconstrained optimization in Chapter 2. Rather than to be exhaustive, it is intended to familiarize the reader with the concepts necessary for the main analysis and shall furthermore motivate the development of our method. In Chapter 3 we formulate SCR and elaborate further on the type of local non-linear models we employ and how these can be solved efficiently. Chapter 4 then gives a detailed theoretical analysis of our method, including the statement of our main convergence theorems. To substantiate the theoretical findings, Chapter 5 summarizes experimental results on well-known machine learning datasets that widely confirm our analysis. Finally, in Chapter 6 we summarize and reflect our results and outline possible future lines of research.

The main contribution of the thesis at hand, namely the theoretical and computational design and analysis of SCR arose in fruitful cooperation with Aurelien Lucchi, which is why the following chapters are not exclusively due to my own work. I am very grateful to my supervisor for his enthusiastic encouragement and assistance as well as numerous interesting research discussions. Many thanks also to Christoph Neumann and Robert Mohr, both PhD candidates at KIT, for sharing their views with me and to Lina

and my friends in Zurich without whom the fall semester would have been as grey as the weather.

Chapter 2

Concepts and Methods for Large Scale Learning

2.1 Notation

Throughout the thesis, scalars are denoted by greek letters, vectors by regular lower case letters and matrices by regular upper case letters. For a vector w, and a matrix A, $\|w\|$ and $\|A\|$ denote the vector ℓ_2-norm and the matrix spectral norm, respectively. $g = \nabla f(w)$ and $H = \nabla^2 f(w)$ are the gradient and the Hessian of f at w. Approximations to H are commonly stated as B. For two symmetric matrices A and B, $A \succeq B$ indicates that $A - B$ is symmetric positive semi-definite. The condition number of a matrix is defined as follows:

Definition 2.1 (Condition number) *Let A be a normal matrix, then its condition number is given by*

$$\kappa(A) = \|A\|/\|A^{-1}\| = \lambda_{\max}(A)/\lambda_{\min}(A), \tag{2.1}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are the maximum and minimum eigenvalues of A.

Regarding the smoothness of the objective functions (and its derivatives) we shall define the following concept:

Definition 2.2 (Lipschitz Continuity) *For $D \subseteq \mathbb{R}^N$ a function $F : D \to \mathbb{R}^m$ is called Lipschitz-continous on D, if*

$$\exists\, L > 0\, \forall w_1, w_2 \in D: \ \|f(w_1) - f(w_2)\| \leq L \cdot \|w_1 - w_2\|$$

.

Our analysis makes extensive use of Taylor's Theorem and the Mean Value Theorem ([Nocedal and Wright, 2006], Chapter 2 and Appendix 2) as well as the Cauchy–Schwarz inequality and triangle inequality ([Conn et al., 2000],

Chapter 2). However, for the sake of brevity, we do not cite these tools each time they are used.

Furthermore, we make use of the following definition of convergences rates:

Definition 2.3 (Convergence Rates) *Let $\{w_k\}$ be a sequence converging to a limit point w^*. We call it*

a) sublinearly convergent, if

$$\exists\, 0 \leq c_k \leq 1 \,\forall\, k \in \mathbb{N} : \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|} \leq c_k, c_k \to 1$$

b) linearly convergent, if

$$\exists\, 0 \leq c \leq 1 \,\forall\, k \in \mathbb{N} : \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|} \leq c$$

c) superlinearly convergent, if

$$\exists\, 0 \leq c_k \leq 1 \,\forall\, k \in \mathbb{N} : \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|} \leq c_k, c_k \to 0$$

d) quadratically convergent, if

$$\exists\, 0 \leq c \leq 1 \,\forall\, k \in \mathbb{N} : \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|^2} \leq c$$

Another way to express these rates is to upper bound the ratio of the distance of the k-th and the 0-th element to w^*. For example one can easily find by recursively applying Definition 2.3.b), that this inequality is equivalent to stating $\|w_k - w^*\| / \|w_0 - w^*\| \leq c^k$ and hence the distance to the limit point at index k is $O(c^k)$. Furthermore, we can write $c^k = e^{k \log(c)} \leq \varepsilon$ to find that a linearly converging sequence achieves an ε distance after $O(ln(1/\varepsilon))$ elements for the first time. Similarly, sublinear convergences is sometimes referred to as $O(1/k)$ and $O(\varepsilon^{-1})$ or even slower: $O(1/\sqrt{k})$ and $O(\varepsilon^{-2})$.

These rates come into play when considering the convergence behaviour of the sequence of iterates generated by optimization algorithms, where the subscript k (e.g. w_k) denotes an iteration counter. We call a method *globally convergent* if its sequence of iterations $\{w_k\}_{k=0}^{\infty}$ converges to a critical point w^* from *any* initialization w_0. Given that a method converges, one can establish upper bounds on the global convergence rate which we shall refer to as *worst case complexity* since these bounds constitute an upper bound on the total number of iterations an algorithm may take to converge globally.

Furthermore, assuming that the trajectory of a method converges to the limit point w^* we are also interested in the *asymptotic* or *local convergence* rate of this method, i.e. the rate of convergence that sets in as soon as some iterate w_k is sufficiently close to w^*.

Regarding the type of limit points that may be approached we make the following distinction:

Definition 2.4 (Critical points) *Let $f : \mathbb{R} \to \mathbb{R}^d$ be twice-differentiable at a point $w^* \in \mathbb{R}^d$. We call w^* a*

a) first-order critical point if and only if $\nabla f(w^*) = 0$.

b) strict saddle point if and only if $\nabla f(w^*) = 0$ and $\nabla^2 f(w^*)$ has at least one positive and one negative eigenvalue.

c) local minimizer or second-order critical point, if $\nabla f(w^*) = 0$ and $\nabla^2 f(w^*) \succeq 0$.

d) global minimizer, if and only if $f(w^*) \leq f(w) \, \forall w \in \mathbb{R}^d$.

Note that the set of saddle points is a subset of the set of critical points. Furthermore, in a convex setting saddle points as in 1.3.a) do not exist and all critical points are local and global minimizers. In a strongly convex setting at most one such critical point exists. Moreover, in both the convex and non-convex setting Definition 1.3.c) of a local minimizer is only necessary and not sufficient for w to be a local minimizer in the sense that there is no other point in the neighbourhood of w that has a strictly lower objective value. A sufficient condition would be to require $\nabla^2 f(w) \succ 0$ but there is a gap between the two since even the strict local minimizer $w = 0$ for $f(w) = w^4$ violates the latter. Consequently, when we talk about local minimizers we specifically also include higher order saddle points where the gradient vanishes but $\nabla^2 f(w) \succeq 0$ and $\exists \, \lambda_n = 0$. Obviously, this consideration is obsolete in the case of strongly convex objectives.

2.2 Problem Description

Common tasks in the field of (supervised) machine learning and statistics are based on identifying patterns or relationships in a set of datapoints $(X, y) \in \mathbb{R}^{n,d} \times \mathbb{R}^d$ where each $x_i \in \mathbb{R}^d$ is an instance with d (nominal or numerical) explanatory variables (*features*) and each y_i is the target variable (*label*) associated with this particular instance. The goal is to find some mapping $h : X \to y$ from the input training set X to the known outcome y which can be nominal (*classification*) or real-valued (*regression*). The mapping itself is commonly scaled by a weight vector $w \in \mathbb{R}^d$ over which the optimization is to be performed.

In this context, it is commonly assumed that the training data are a statistical sample drawn independently from some fixed, unknown probability distribution $(X, y) \sim \mathcal{D}$, which allows us to interpret the learning problem as a problem of statistical inference[1]. Given the data, one then chooses a specific family of functions $\mathcal{H} := \{h(X, w)\}$ (i.e. a bag of possible models) and tries to find the one hypothesis $h(X, w^*)$ out of this hypothesis space that on average has the lowest error of prediction $l(h(X, w), y)$ across *all* possible realizations of the data, particularly including unobservable realizations.

Towards this goal a natural approach would be to minimize the so-called *expected risk* in order to find the parameters that give rise to the desired hypothesis

$$w^* = \arg\min_{w \in \mathbb{R}} \int l(h(X, w), y) dD(X, y) = \arg\min_{w \in \mathbb{R}} \mathbb{E}\left[l(h(X, w), y)\right]_{\mathcal{D}}. \quad (2.2)$$

However, since the generative process of X and y is unknown we cannot form the expected value with respect to $\mathcal{D}$. Instead we minimize the so-called *empirical risk* with respect to the observed samples in the hope of approximating the expected risk, i.e. retaining a high predictive accuracy on *unseen* data

$$\hat{w} = \arg\min_{w \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} l(h(x_i, w), y_i). \quad (2.3)$$

Fortunately, based on the pioneering work on uniform convergence by Vapnik and Chervonenkis [1971], bounds of the following type can be shown to hold

$$\mathbb{E}\left[l(h(X, \hat{w}), y)\right]_{\mathcal{D}} \leq \mathbb{E}\left[l(h(X, w^*), y)\right]_{\mathcal{D}} + O\left(\sqrt{\frac{VC(\mathcal{H})}{n}}\right) \quad w.h.p. \quad (2.4)$$

In this regard the so called VC dimension $VC(\mathcal{H})$ can be interpreted as a measure of model complexity and is assumed to be finite. As a result, the optimality gap between the empirical and expected loss minimizer increases when a richer family of prediction functions is employed (due to the risk of overfitting). Furthermore, for a fixed hypothesis space $\mathcal{H}$ the generalization error of the hypothesis $h(X, \hat{w})$ is within $O(\sqrt{1/n})$ of that of the true population risk minimizer $h(X, w^*)$ and thus inversely proportional to the number of data points in X. Consequently, in the large scale learning setting that we consider in this thesis where $n \gg 1$ minimizing (2.3) is reasonably likely to yield a low generalization error.

[1] The joint probability distribution $\mathcal{D}(w, y)$ represents both, the distribution of features $D(w)$ as well as the conditional label probability $D(y|X)$.

To simplify matters we shall define the function f as a composition of the loss function l and the prediction h

$$f_i(w) = l(h(x_i, w), y). \tag{2.5}$$

This gives rise to the empirical risk minimization problem as stated in (1.1) that represents an unconstrained, continuous and deterministic optimization problem. At this point we remark that often an (easily to compute) regularization term $r(w)$ parametrized by a scalar $\lambda > 0$ is added to the objective and that the methods discussed in the subsequent sections can be applied readily to the task of minimizing the *regularized empirical loss* as long as r is smooth.

Finally, throughout the entire analysis we assume that the objective is smooth and twice continuously differentiable $f \in C^2$ and note that the sum-structure of the objective is obviously also inherited by the first and second derivatives of f and we can thus write:

$$\nabla f = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i \text{ and } \nabla^2 f = \frac{1}{n} \sum_{i=1}^{n} \nabla^2 f_i \tag{2.6}$$

Popular examples of machine learning algorithms that include minimizing (1.1) are linear regression (squared loss), logistic regression (log loss) and support vector machines (hinge loss) which all three give rise to a convex problem setting. Yet, non-convex problems also frequently arise in machine learning like for example dictionary learning, gaussian mixture models and training neural networks. For this kind of problems there is sound evidence from statistical physics and random matrix theory that within the group of critical points, saddle points prevail in high dimensions, especially in areas where the error is relatively high (Bray and Dean [2007]). These results are in line with the findings of Choromanska et al. [2015]. They specifically investigate the error surface of multilayer neural networks and find that the objective values of local minimizer are located in a certain band lower-bounded by the global minimum and that above this band the number of local minima decreases exponentially in the size of the network while the occurrence of saddle points accumulates.

As laid out in the discussion following Definition (2.4) also higher order saddles points like the well-known monkey saddle may arise in a non-convex problem. While it can be shown theoretically that the subset of functions with exclusively strict critical points is open and dense in $C^2(\mathbb{R}^n, \mathbb{R}^n)$ (Jongen et al. [2013]), real world applications may still give rise to high order saddle points e.g. due to over-specified models or permutation-symmetry in multi-layer neural nets. Since one needs third order information in order to escape such saddles we will not differentiate between *true* and degenerate

local minimizers in this thesis. The reader is referred to the work of Anandkumar and Ge [2016] for such a method. Furthermore, some problems like dictionary learning and orthogonal tensor decomposition have indeed been proven to have a strict saddle structure such that the Hessian of every saddle point has a negative eigenvalue and thus there always exists direction of negative curvature along which second-order methods can progress (Sun et al. [2015]).

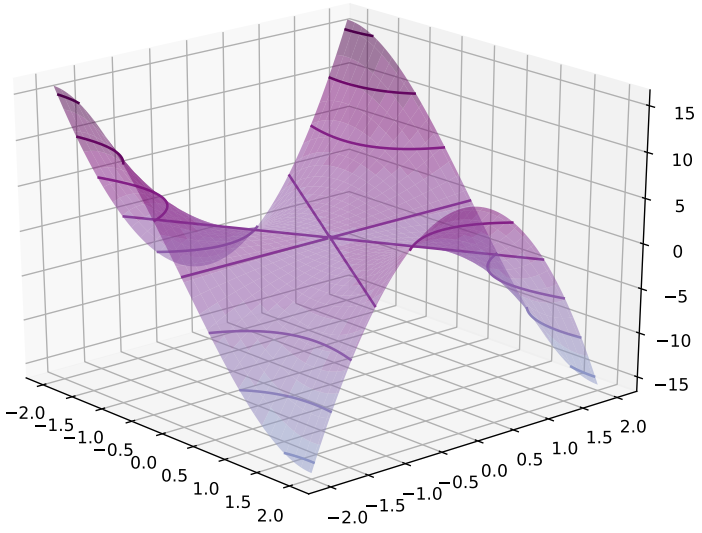


Figure 2.1: Graph of the function $w_1^3 - 3w_1w_2^2$ which gives rise to the monkey saddle

Finally, we note that finding a global minimizer can be extremely hard. As a matter of fact, the work of Hillar and 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 in the hope that this optimum yields a comparably low objective value.

2.3 Overview of Common Methods

For the sake of completeness, let us now review some fundamental optimization algorithms that can be applied to solve (1.1). We explicitly consider only first and second-order methods with updates of the form

$$w_{k+1} = w_k - \eta_k Q_k \nabla f(w_k), \tag{2.7}$$

where η_k is the learning rate of iteration k and Q_k is a suitable scaling matrix that may provide curvature information. Derivative free metaheuristics like

random search or particle swarm optimization are not treated in this thesis.

First-Order Methods The prototypical optimization method is Gradient Descent (**GD**) which, based on the fact that the gradient $\nabla f(w)$ points into the direction of steepest *ascent*, takes a step along the *negative* gradient with a learning rate $\eta_k > 0$. Hence the scaling matrix Q_k equals the identity matrix I, the updates become

$$w_{k+1} = w_k - \eta_k \nabla f(w_k) \quad \text{(GD)} \tag{2.8}$$

and the per-iteration cost boils down to evaluating n gradients $O(nd)$. GD with a constant learning rate is converging to a first-order critical points at a sub-linear rate $O(1/k)$ if f is convex and at a linear rate $O(\rho^k)$ if f is strongly convex. However, note that the constant $\rho = 1 - (1/\kappa(H^*))$ gets worse the higher the condition number of the Hessian at the limit point w^* is ([Nesterov, 2013]).

An increasingly popular variant called Stochastic Gradient Descent (**SGD**) reduces the per-iteration cost to $O(d)$ by using only one datapoint at a time to compute the update

$$w_{k+1} = w_k - \eta_k \nabla f_i(w_k) \quad \text{(SGD)}. \tag{2.9}$$

Of course, the iterates $\{w_k\}$ generated by SGD now constitute a random sequence that depends on the sequence of sampled datapoints $\{i_k\}$ and the objective value is no longer guaranteed to decrease in each iteration. However, when sampled uniformly and independently, the direction $-\nabla f_i$ is a descent direction *in expectation*. Furthermore, a decreasing learning rate is needed for this method to converge and the convergence rate reduces (compared to GD) to the sub-linear levels $O(1/\sqrt{k})$ in a convex and $O(1/k)$ in a strongly convex setting[Nemirovski et al., 2009]. Yet, the cheaper per-iteration cost may overcompensate the slower convergence rate in terms of total computation time when a large number of datapoints is involved[2].

Variance Reduced (**VR**) variants make explicit use of the finite sum-structure in (1.1) in order to retain GDs linear convergence rate while keeping the learning rate constant[3]. The basic idea is the following: By sampling one gradient $\nabla f_i(w_k)$ per-iteration we try to estimate its expected value, the full gradient $\nabla f(w_k)$, cheaply. Given that some prior information, say $\nabla f_i(\alpha_i)$ and $\nabla f(\alpha) = \frac{1}{n}\sum_j^n \nabla f_j(\alpha)$, has been stored we can estimate $\nabla f(w_k)$ instead by $\theta_k = \nabla f_i(w_k) - \nabla f_i(\alpha_i) + \nabla f(\alpha)$. Given that $\mathbb{E}[\nabla f_i(\alpha_i)] = \nabla f(\alpha)$ it

[2] compare the total times $O(nd) * O(\log(1/\varepsilon))$ of GD to $O(d) * O(1/\varepsilon)$ of SGD which are needed to achieve ε optimality asymptomatically.

[3] Johnson and Zhang [2013] show that the variance of SGD can only go to zero if a decreasing learning rate is used.

becomes evident that θ_k is still an unbiased estimator, however the variance $\text{Var}(\theta_k) = \text{Var}(\nabla f_i(w_k)) + \text{Var}(\nabla f_i(\alpha_i)) - 2\,\text{Cov}(\nabla f_i(w_k), \nabla f_i(\alpha_i))$ is reduced given that the covariance term is large enough. Hence the iterate update scheme is written as follows:

$$w_{k+1} = w_k - \eta_k \left(\nabla f_i(w_k) - \nabla f_i(\alpha_i) + \frac{1}{n} \sum_j^n \nabla f_j(\alpha_j) \right) \quad \text{(VR)} \tag{2.10}$$

which generally has a runtime of $O(d)$ but comes at the cost of needing to store past information. Different methods use different techniques to obtain α_i which is furthermore updated regularly in order to keep the covariance high. Two particularly popular approaches are SVRG, which sets $\alpha_i = w_l, \forall i$ and hence computes and stores the *full* gradient ($O(nd)$ computation and $O(d)$ storage cost) every other iteration (Johnson and Zhang [2013]) and SAGA which stores the initial gradient of *each* datapoint in a table $\alpha_i = w_0, \forall i$ ($O(nd)$ storage cost) and updates only *one* entry in every iteration to the most recent iterate $\alpha_i = w_k, i = sampled\,index$ (Defazio et al. [2014]).

As all first-order methods only use gradient information, there is no global convergence guarantee towards a second-order local minimizer in the general setting of non-convex function.

Second-Order Methods The canonical second-order method is Newton's methods (**NM**) which uses the inverse Hessian as a scaling matrix $Q_K = \nabla^2 f(w_k)^{-1} = H(w_k)^{-1}$ and thus has updates of the form

$$w_{k+1} = w_k - H(w_k)^{-1} \nabla f(w_k) \quad \text{(NM)}. \tag{2.11}$$

Using curvature information to rescale the steepest descent direction gives Newton's method the useful property of being linearly scale invariant, i.e. for some (non-singular) re-parametrization $\tilde{w} = Aw$ the optimal update given by this method remains $\tilde{w}_{k+1} = \tilde{w}_k - H(Aw_k)^{-1} \nabla f(Aw_k) = \tilde{w}_k - H(\tilde{w}_k)^{-1} \nabla f(\tilde{w}_k)$. Contrary to the above presented first-order methods, this gives rise to a *problem independent* local convergence rate that is furthermore also faster, namely super-linear and even quadratic in the case of Lipschitz continuous Hessians (see Nocedal and Wright [2006] Theorem 3.5).

However, there are certain drawbacks about applying classical Newton's method. First of all, the Hessian matrix may be singular and thus not invertible. Secondly, even if it is invertible the Newton direction is not necessarily a direction of descent and hence arbitrary critical points (including local maxima) may be approached. Finally, the cost of forming and inverting the Hessian sum up to $O(nd^2 + d^3)$ and are thus prohibitively high for applications in high dimensional problems. To tackle the first and second issue sophisticated line search approaches modify H to make it positive definite

for example by adding a positive diagonal matrix $B = H + \lambda I$ with some $\lambda > 0$. Alternatively, trust region methods can be used to make Newton's method globally convergent which we will present in great detail later.

The last issue can be partially resolved by applying so-called quasi-Newton methods like the celebrated Broyden-Fletcher-Goldfarb-Shanno (**BFGS**) algorithm which construct the matrix Q_k in a computationally more feasible way ($O(nd + d^2)$) while preserving sufficient second-order information. Towards this end the change in the gradients and iterates from one iteration to another is used to dynamically update a direct approximation of the *inverse* H_k^{-1}. Specially, the BFGS methods computes $s_k = w_{k+1} - w_k$ and $v_k = g_{k+1} - g_k$, sets

$$Q_k = \left(I - \frac{s_{k-1}v_{k-1}^\intercal}{s_{k-1}^\intercal v_{k-1}}\right)^\intercal Q_{k-1} \left(I - \frac{v_{k-1}s_{k-1}^\intercal}{s_{k-1}^\intercal v_{k-1}}\right) + \frac{s_{k-1}s_{k-1}^\intercal}{s_{k-1}^\intercal v_{k-1}} \tag{2.12}$$

and then updates the current iterate according to the following scheme

$$w_{k+1} = w_k - Q(w_k)^{-1}\nabla f(w_k) \quad \text{(BFGS)}. \tag{2.13}$$

Notably, the updated approximations Q_k from (2.12) are symmetric, positive definite and satisfy the secant equation $Q_k v_{k-1} = s_{k-1}$.

While this routine also yields a superlinear convergence rate the construction of Q as in (2.12) has the downside of producing dense scaling matrices even when the exact Hessian is sparse which limits the approach to application in small and midsize problems ($O(d^2)$ storing cost) [Nocedal and Wright, 2006] Chapter 6 . A common alternative is the so-called *limited memory* BFGS or L-BFGS. This method does not form Q_k explicitly but directly computes the matrix-vector product $Q_k^{-1}\nabla f(w_k)$ based on the last $m > 0$ displacement pairs (s, v) that have been stored in memory. While this approach incurs per-iteration cost in the order of $O(md)$, only a linear local convergence rate can be proven [Liu and Nocedal, 1989].

Of course, it is possible to apply the variance reduction techniques presented above within a quasi-Newton framework and this combination may lead to practical improvements as demonstrated by Lucchi et al. [2015].

Sub-sampled second-order Methods Based on the pioneering work of Byrd et al. [2011], constructing the scaling matrix Q_k by usage of *sub-sampling* techniques for the Hessian has become an increasingly popular alternative to quasi-Newton methods (see e.g. Erdogdu and Montanari [2015], Roosta-Khorasani and Mahoney [2016a] and Agarwal et al. [2016b]). The intuition behind this approach is that in many large-scale applications the data does involve a good deal of (approximate) redundancy which makes using all of the samples in every iteration computationally inefficient. Thus, in each

iteration k sub-sampled Newton methods choose a subset $\mathcal{S}_H \in \{1, 2, \ldots, n\}$ at random form

$$Q_k = \frac{1}{n} \sum_{i \in \mathcal{S}} H_i(w_k). \tag{2.14}$$

Assuming that the sub-sampled Hessian Q_k is invertible the update can be written as follows

$$w_{k+1} = w_k - \eta_k Q_k^{-1} \nabla f(w_k) \quad \text{(subNewton)}. \tag{2.15}$$

Of course gradient and Hessian information may also be sub-sampled at the same time which gives a fully stochastic algorithm. Generally it has been shown that sub-sampled Newton methods can be made globally converging on strongly convex objectives. Furthermore, linear convergence can be achieved by uniformly sampling the Hessian and it is sampled to an increasing accuracy as the algorithm progresses even superlinear can be obtained Roosta-Khorasani and Mahoney [2016b]. Remarkably, Martens [2010] was the first to apply a sub-sampled hessian-free damped newton method with increasing sample size for deep learning and was able to (empirically) improve on the commonly observed under-fitting behaviour of SGD in the context of training deep neural networks. Since the damped Newton approach puts a quadratic penalty on the step size this work is fairly close to our approach and thus suggests its application on neural networks. As stated above these networks give rise to many high level saddle points which can be efficiently escaped using second-order information as we shall elaborate in the next section.

Finally we want to note that all of the large majority of recently proposed sub-sampled Newton methods are line-search type methods and the only stochastic trust region approaches that we are aware of are those of Martens [2010] and Blanchet et al. [2016]. As discussed, the former applies only a very minimalist TR framework and furthermore gives no theoretical convergence analysis at all. The latter analyses the first-order global convergence of a *first-order* trust region method on non-convex functions based on the properties of supermartingales. Hence our approach is to the best of our knowledge the first to explore Hessian sub-sampling in a trust region and specifically in a cubic regularization framework and also the first to give a quadratic convergence rate as well as a second-order guarantee of such a method on non-convex objectives.

2.4 Why second-order?

Fast asymptotic convergence and highly accurate solutions As we have seen above the local convergence rate of first-order methods suffers from a high condition number of the Hessian. Geometrically, this term can be

interpreted as the maximum stretch of the unit sphere by H, where the unit sphere of a vector space V in a certain norm $\|\cdot\|$ is the set of all points w of distance 1 from its central point ($\{w \in V : \|w\| = 1\}$). In other words, a large discrepancy between $\lambda_{min}(H)$ and $\lambda_{max}(H)$ causes an equally large discrepancy between the longest and the shortest half-axis of the resulting ellipsoid which generally leads to more elongated level curves $lev_f^\alpha = \{w \in \mathbb{R}^n | q(w) = \alpha\}$ of an objective near to its minimizer[4]. This, in turn, causes gradient descent methods to enter the so-called *zig-zagging* behaviour which may lead to unacceptably slow asymptotic convergence rates even for strongly convex objectives.

Figure 2.2 illustrates this effect for a convex-quadratic objective $w^\intercal w, w \in \mathbb{R}^2$ before (left) and after the linear transformation $A = diag(1, \sqrt{5})$. As can be seen, Newton's method finds the global minimizer of the quadratic objective within one step in both cases due to the discussed linear scale invariance.

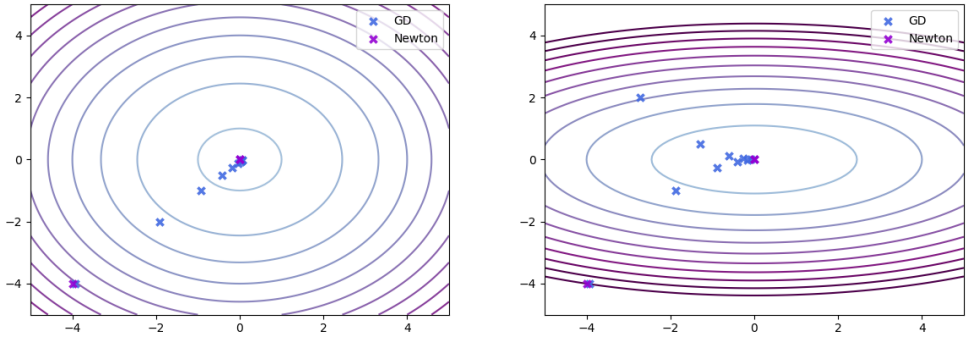


Figure 2.2: Level sets of a convex quadratic objective and iterates of Gradient Descent and Newton's method

But why does rescaling the gradient by curvature information give rise to more sensible directions? Intuitively, the curvature in a particular descent direction d signals how much the gradient of the objective changes along this direction. Thus, in case of low (positive) curvature, the gradient will change slowly along d (i.e. d stays a descent direction over a large range) and hence it is appropriate to take a step s that moves far along d (i.e. by making $s^\intercal d$ large), even if the first-order information itself signals a relatively small objective decrease, i.e. $-\nabla f^\intercal s$ is small. Following the same arguments it is thus advisable to choose a step that does not go far along d in case of high curvature in this direction. As a consequence, the zig-zagging

[4] In practice it is not uncommon to encounter problems with $\kappa(H)$ in the order of 10^3 to 10^6 (see Section 5)

behaviour of GD can be attributed to repeatedly taking steps that travel too far in directions of high curvature. Newton's method, on the other hand, computes the distance to go along the gradient as the ratio of reduction indicated by the steepness divided by the associated curvature information: $-\|\nabla f\|/(\nabla f^\intercal H \nabla f)$. This is precisely the step size after which the objective f increases in the case that it is a convex-quadratic function.

As a result the Newton step can be interpreted as the step to the critical point of the following local quadratic approximation of f around the current iterate w:

$$m(s) = f(w) + \nabla f(w)^\intercal s + \frac{1}{2} s^\intercal H s \tag{2.16}$$

which can be shown to be $O(\|s\|^3)$ close to f based on a second-order Taylor approximation.

It is sometimes argued that one never really looks for highly accurate solutions in Machine Learning due to the risk of over-fitting the training dataset. In this context the asymptotic properties of steepest descent methods are then considered a feature rather than a bug. However, we have already laid out in equation (2.4) that there is no point in settling with an inaccurate solution to the empirical risk minimization when n is high because the risk of over-fitting vanishes[5]. Furthermore, in some applications like Generalized Linear Models the learned vector w contains specific meanings that one might be interested in interpreting for the sake of understanding real world relationships.

Escaping saddles quickly A further disadvantage of first-order methods is that there is no global convergence guarantee regarding second-order critical points which may cause them to get stuck at a saddle point with an arbitrarily bad objective value. The classic Newton method doesn't provide such a guarantee either but it can be globalized easily as mentioned in Section 2.3.

To be fair, though, one has to admit that it is actually highly unlikely for a gradient descent method to converge to a strict saddle. As a matter of fact Panageas and Piliouras [2016] show that, if initialized at random, GD does not converge to a saddle point with probability one. Consider for example the function $f(w_1, w_2) = 1/2w_1^2 + 1/4w_2^4 - 1/2w_2^2$ which is non-convex but coercive and has global minimizers at $\bar{w} = (0, 1)$ and $w^* = (0, -1)$. If starting anywhere along the line segment $(w_1, 0)$, the steepest descent direction $-\nabla f(w_1, 0) = -(w_1, 0)$ points right at the saddle $\hat{w} = (0, 0)$ and GD will end up getting stuck there. However, from *any* other initial point it would surely find one of the global minimizers. Stochastic variants like SGD that use a noisy gradient approximation for their step calculation are even less

[5] given that one has opted for a suitable model in the first place. See Eq. (2.2)

likely to halt at the saddle. As a matter of fact, Ge et al. [2015] prove that the convergence rate of a (noise injected) SGD algorithm has a polynomial dependency on the inverse of the gradient norms around the saddle.

Nevertheless, a closer look at Figure 2.3 a) suggests that saddles delay the progress of first-order methods significantly since directions of low curvature are explored with too small steps. Furthermore, it illustrates that NM, even though it was initialized off the saddle convergence line $(w_1, 0)$, is indeed attracted by first-order critical points. Yet, as can be seen in Figure 2.3 b) this issue as well as the small progress problem are resolved when a trust region framework is applied[6].

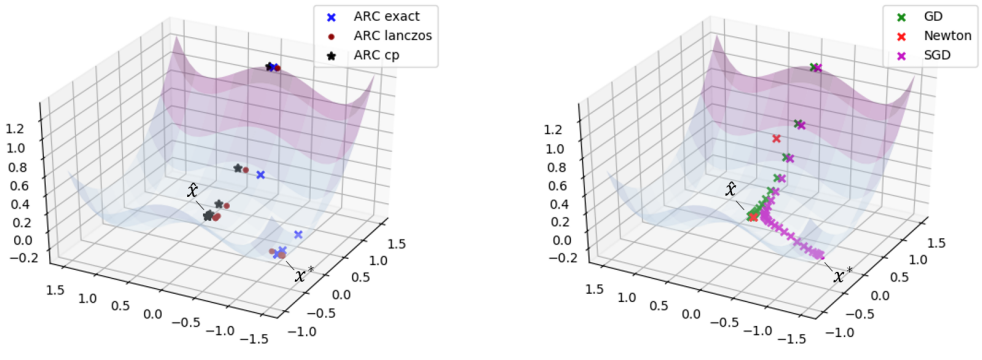


Figure 2.3: Graph of the function $1/2w_1^2 + 1/4w_2^4 - 1/2w_2^2$ and the iterates of different first- and second-order optimization methods

All in all, the ubiquity of high error saddle points in high dimensional problems as introduced in Section 2.2 seems to make it particularly hard for curvature-free methods like gradient-descent to quickly progress towards regions of lower loss. This impedes rapid high dimensional non-convex optimization with first-order methods and thus strongly suggests the use of second-order information.

2.5 Adaptive Cubic Regularization

Trust Region As we have just seen it can be of great advantage to take second-order information into consideration when minimizing both: con-

[6] ARC exact (blue) minimizes the subproblems globally while ARC lanczos (red) uses does so in Krylov subspaces. ARC cp (black) takes Cauchy steps which are not enough for second-order convergence. Details follow in Section 3.2.

vex and non-convex functions. However, next to the drawbacks of Newton's method discussed in Section 2.3 (namely that steps may be ascending or not even computable) another issue is that the local quadratic model (2.17) that is minimized in each NM iteration may simply be an in-adequate approximation of the true objective. A powerful framework to resolve both of these issues is the so-called trust region approach (TR). These methods also construct a quadratic model m_k but constrain the subproblem in such a way that the stepsize is restricted to stay within a certain radius Δ_k within which the model is trusted to be sufficiently adequate:

$$\min_{s\in\mathbb{R}^d} m_k(s) = f(w_k) + \nabla f(w_k)^\intercal s + \frac{1}{2} s^\intercal H(w_k) s, \;\; s.t. \; \|s\| \leq \Delta_k \quad \text{(TR)} \tag{2.17}$$

Hence, contrary to line-search methods this approach finds the step s_k and its length $\|s_k\|$ *simultaneously* by minimizing (2.17). Subsequently the function decrease $f(w_k) - f(w_k + s_k)$ is compared to the model (or *predicted*) decrease $m(0) - m(s_k)$ and the step is only accepted if this ratio ρ exceeds some predefined success threshold. As a consequence, the Newton Step $s^N = -H_k^{-1} g_k$ is only taken if it lies within the trust region radius and yields a certain amount of decrease in the objective value. Finally, the trust radius is updated adaptively depending on ρ_k as it constitutes a measure for the model adequacy. Since many functions look somehow quadratic close to a minimizer the radius can be expected to grow asymptotically such that eventually full Newton steps are taken in every iteration which retains the local quadratic convergence rate.

However, when not in a neighbourhood of the minimizer, the issue with quadratic approximations is that they assume a constant curvature and may thus become inadequate quickly for general non-linear functions with varying curvature. Let us take another look at the saddle function introduced in Figure 2.3. Obviously, slicing the error surface along the w_1 axis gives rise to a parabola but a saddle arises due to the order 4 polynomial in w_2. This can also be seen when looking at the Hessian $H(w) = diag(1, 3w_2^2 - 1)$.

As can be seen on the left hand side of Figure 2.4 the convex-quadratic approximation $m(s)$ around $\bar{w}$ drastically underestimates the objective $f(w)$ for points w with $w_2 > \bar{w}_2$ because of the rapid curvature change in this direction. As a consequence, Newton's method (purple) initialized at $\bar{w}$ takes a step that travels too long and ends up on a higher level set than where it started, as can be seen on the right. The trust region method (shown in blue for different values of $\Delta \in [0, 1.25]$) does far better by altering the step directions towards the minimizer and furthermore rejecting any step that travels further than $\|s\| = 1.25$ even before the function decrease is negative because of a too low decrease ratio ρ.

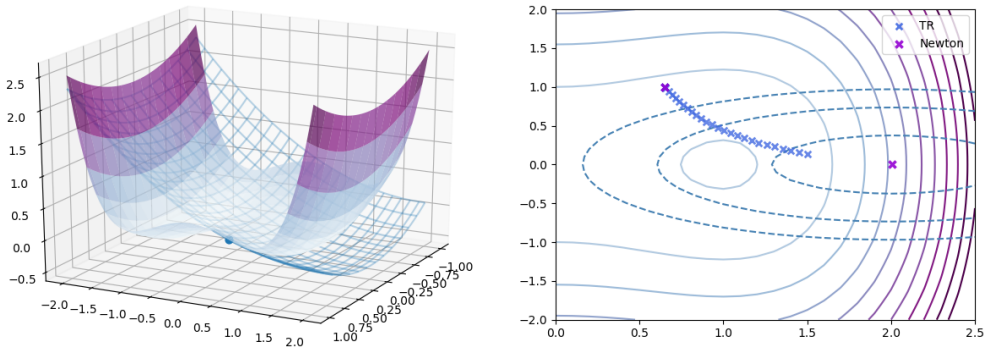


Figure 2.4: Graph and level sets of the function $f(w) = 1/2w_1^2 + 1/4w_2^4 - 1/2w_2^2$ and a local quadratic approximation $m(\bar{w}, s)$ formed around the point $\bar{w} = (1, \sqrt{3} + 0.01)$

Nesterov 2006 In a Lagrangian manner the constrained trust region solution can (in the case of boundary solutions) be shown to be equivalent to minimizing the model function with a suitable *quadratic* penalty term on the stepsize (see Proof following Theorem 7.2.1 in Conn et al. [2000]). Thus, a natural extension is the cubic regularization method introduced by Nesterov and Polyak [2006] which use a *cubic* penalty term as a regularization technique for the computation of the trial step. Assuming a Lipschitz continuous Hessian with constant κ_H and setting the penalty parameter to this constant in each iteration they show that the model

$$m_k(s_k) := f(w_k) + s_k^\intercal \nabla f(w_k) + \frac{1}{2} s_k^\intercal H_k s_k + \frac{\kappa_H}{6} \|s_k\|^3 \quad \text{(Nesterov 2006)} \quad (2.18)$$

is a global over-estimator of the objective, i.e. $f(w_k + s_k) \leq m(s_k)\ \forall w, s \in \mathbb{R}^d$ and hence any minimizer of m_k is guaranteed to yield a positive function decrease in f. Nesterov and Polyak [2006] were able to show that, if the step is computed by globally minimizing the cubic model, this method possesses the best known worst case complexity to solve Eq. (1.1): an overall worst-case iteration count of order $\varepsilon^{-3/2}$ for generating $\|\nabla f(w_k)\| \leq \varepsilon$, and of order ε^{-3} for achieving approximate non-negative curvature. Consider Table 2.1 for a comparison with other methods[7].

ARC However, minimizing (2.18) in an exact manner impedes the performance of this method for large scale learning applications as it requires

[7] Interestingly, Cartis et al. [2012] show that the first-order worst case complexity bounds are sharp which reveals the fact that the classical NM may be as slow as GD!

Method	**local rate**	$\|\nabla f\| \leq \varepsilon$	$\nabla^2 f \succeq -\varepsilon$	**Source**
GD	linear	$O\left(\frac{nd}{\varepsilon^2}\right)$	n/a	[Cartis et al., 2010]
SGD	sublinear	$O\left(\frac{d}{\varepsilon^4}\right)$	n/a	[Ghadimi and Lan, 2016]
SVRG	linear	$O\left(nd + \frac{n^{2/3}d}{\varepsilon^2}\right)$	n/a	[Allen-Zhu and Hazan, 2016]
NM	superlinear	$O\left(\frac{nd^2 + d^3}{\varepsilon^2}\right)$	n/a	[Cartis et al., 2010]
quasi NMs	superlinear	n/a	n/a	[Nocedal and Wright, 2006]
sub NMs	(super)linear	n/a	n/a	[Erdogdu and Montanari, 2015]
TR	superlinear	$O\left(\frac{nd^2}{\varepsilon^2}\right)$	$O\left(\frac{nd^2}{\varepsilon^3}\right)$	[Gratton et al., 2008]
ARC	superlinear	$O\left(\frac{nd^2}{\varepsilon^{3/2}}\right)$	$O\left(\frac{nd^2}{\varepsilon^3}\right)$	[Cartis et al., 2010]

Table 2.1: Comparison of common optimization methods in terms of local convergence rate and worst case time complexity for achieving ε first and second-order criticality.

access to the full Hessian matrix. More recently, Cartis et al. [2011] presented a method (hereafter referred to as ARC) which introduced three crucial changes to make the method more practical. First, they relaxed the Lipschitz continuity assumption of the Hessian and introduced an adaptive penalty parameter σ_k which is updated in the spirit of trust region methods. Second, they derived a condition that allows the use of a quadratic approximation B_k that is sufficiently close to H_k in the following way:

$$\|(B_k - H_k)s_k\| \leq C\|s_k\|^2, \forall k \geq 0, C > 0 \tag{2.19}$$

Finally, they showed that it is sufficient to find an *approximate* subproblem minimizer e.g. by applying a Lanczos-type method to build up evolving Krylov spaces, which can be constructed in a Hessian-free manner (i.e. by accessing the Hessian only indirectly via matrix-vector products). Thus the ARC method minimizes subproblems of the following type at each iteration k:

$$m_k(s_k) := f(w_k) + s_k^\intercal \nabla f(w_k) + \frac{1}{2} s_k^\intercal B_k s_k + \frac{\sigma_k}{3}\|s_k\|^3 \quad \text{(ARC)} \tag{2.20}$$

where $\sigma_k > 0$.

However, there are still three major obstacles for the application of ARC in the field of machine learning: (1) The cost of the Lanczos process increases linearly in n and can thus become very expensive for large datasets, (2) the use of exact gradient information impedes applications where n is so large

that even computing the full gradient is too expensive and (3) there is no theoretical guarantee that quasi-Newton approximations such as (2.12) satisfy Eq. (2.19) and Cartis et al. [2011] do not provide any alternative approximation technique.

In this work we set out to resolve these issues in order to come up with a computationally efficient cubic regularization variant that retains all of ARCs outstanding global and local convergence properties. In this regard, we make the following contributions

- Based on concentration inequality bounds we provide a theoretical Hessian sampling scheme that is guaranteed to satisfy Eq. (2.19) with high probability.
- 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.
- We extend the analysis to inexact gradients and prove that the convergence guarantees of Nesterov and Polyak [2006], Cartis et al. [2011] can be retained.
- Finally, to substantiate the theoretical findings, we provide experimental results demonstrating significant speed-ups compared to standard first and second-order optimization methods for various convex and non-convex objectives.

Chapter 3

Stochastic Cubic Regularization

3.1 Formulation

As laid out above, we are interested in optimizing potentially (non-convex) sum-structured empirical risk minimization problems such as Eq. (1.1) in a large-scale setting where the number of datapoints is very large and the dimensionality of the problem is large ($n \gg d \gg 1$), such that the cost of solving the cubic regularization subproblem Eq. (2.20) 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 convergence guarantees as well as worst-case complexity bounds. A detailed theoretical analysis is given in Chapter 4. Here we shall first state the algorithm itself and elaborate further on the type of local non-linear models we employ and how these can be solved efficiently.

Model objective Instead of using deterministic gradient and Hessian information as in Eq. (2.20) in every iteration k we apply unbiased estimates of these quantities constructed from two independent subsets of points denoted by $\mathcal{S}_g$ and $\mathcal{S}_B$ which represent a collection of unique indices from $\{1, 2, \ldots, n\}$ and whose cardinality is denoted by $|\mathcal{S}|$, with $0 < |\mathcal{S}| \leq n$. We then construct a local cubic model that is (approximately) minimized in each iteration:

$$m_k(s) := f(w_k) + s^\intercal g_k + \frac{1}{2} s^\intercal B_k s + \frac{\sigma_k}{3} \|s\|^3 \tag{3.1}$$

where $g_k := \frac{1}{|\mathcal{S}_g|} \sum_{i \in \mathcal{S}_g} \nabla f_i(w_k)$ and $B_k := \frac{1}{|\mathcal{S}_B|} \sum_{i \in \mathcal{S}_B} \nabla^2 f_i(w_k)$.

Note that this objective may be non-convex and may have global and local minimizers as well as saddle points, as can be seen in Figure 3.1. Its derivative with respect to s_k is defined as

$$\nabla m_k(s) = g_k + B_k s + \lambda s \tag{3.2}$$

and hence the second derivative can be written as follows

$$\nabla^2 m_k(s) = B_k + \lambda I + \lambda \left(\frac{s}{\|s\|}\right)\left(\frac{s}{\|s\|}\right)^\intercal, \tag{3.3}$$

where $\lambda = \sigma_k \|s\|$. This term can be interpreted as a damping parameter that re-conditions the Hessian to be positive semidefinite as done for example in the Levenbeg-Marquard method [Nocedal and Wright, 2006]. Note that the last summand in (3.3) is merely a constant independent of s for *quadratically* regularized methods such as trust region algorithms.

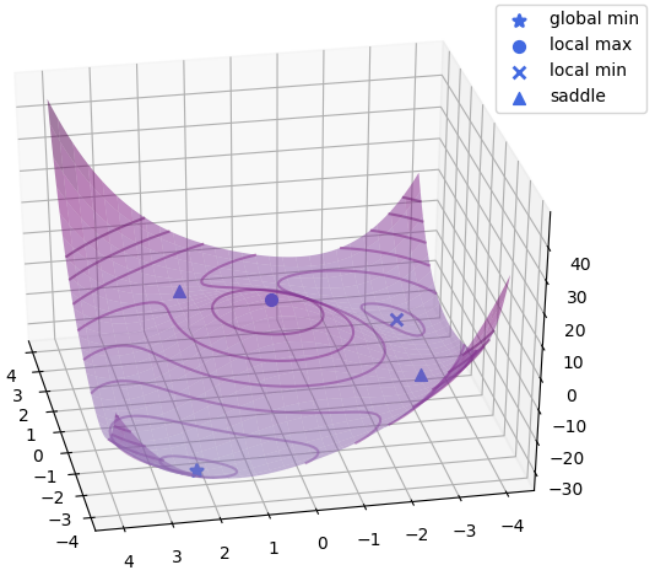


Figure 3.1: Graph of a two-dimensional instance of (3.7) with $g = (1/2, 0)^\intercal$ and $H = [-10, 1; 1, -10]$

Algorithm Our Stochastic Cubic Regularization (SCR) method is presented in Algorithm 1. At iteration k, we sub-sample two sets of datapoints uniformly and independently from which we compute a stochastic estimate of the gradient and the Hessian. We then minimize the obejctive from Eq. (3.7) with respect to s *approximately* using the method described in Section 3.2.3 to obtain a trial step which is only accepted if the function decrease is greater or equal to $\eta_1 \cdot 100\%$ of the model decrease. Finally, we update the regularization parameter σ_k also depending on how well the model approximates the true 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. Inspired by Cartis et al. [2011] the intention here is to reduce the penalty rapidly to al-

Algorithm 1 Sub-sampled Cubic Regularization (SCR)

Input:
Starting point $w_0 \in \mathbb{R}^d$ (e.g $w_0 = \mathbf{0}$)
$\gamma > 1, 1 > \eta_2 > \eta_1 > 0$, and $\sigma_0 > 0$
for $k = 0, 1, \ldots,$ until convergence **do**
Sample gradient g_k and Hessian H_k according to Eq. (4.26) and Eq. (4.32) respectively
Obtain s_k by solving $m_k(s)$ (Eq. (3.7)) such that A3.7 holds
Compute $f(w_k + s_k)$ and

$$\rho_k = \frac{f(w_k) - f(w_k + s_k)}{f(w_k) - m_k(s_k)} \tag{3.4}$$

Set

$$w_{k+1} = \begin{cases} w_k + s_k & \text{if } \rho_k \geq \eta_1 \\ w_k & \text{otherwise} \end{cases} \tag{3.5}$$

Set

$$\sigma_{k+1} = \begin{cases} \max\{\min\{\sigma_k, \|g_k\|\}, \varepsilon_m\} & \text{if } \rho_k > \eta_2 \text{ (very successful iteration)} \\ \sigma_k & \text{if } \eta_2 \geq \rho_k \geq \eta_1 \text{ (successful iteration)} \\ \gamma\sigma_k & \text{otherwise (unsuccessful iteration),} \end{cases} \tag{3.6}$$

where $\varepsilon_m \approx 10^{-16}$ is the relative machine precision.

low for (almost[1]) Newton steps once convergence sets in, while preserving some regularisation non-asymptotically. In case of unsuccessful iterations we reject the trial step and increase the penalty parameter in order to reduce the length of the next trial step to a region where the quadratic approximation is more likely to be accurate.

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. Note that contrary to the deterministic case (ARC), the successfulness of an SCR iteration now depends on two issues: a) the adequacy of a quadratic model in itself *and* b) the accuracy of the sub-sampled quantities. We thus re-sample after both, successful and unsuccessful iteration and note that due to the decrease in stepsize the sampling schemes (4.26) and (4.32) increase $|S|$ after unsuccessful iterations[2].

[1]Note that contrary to the trust region approach, cubic regularization methods never take full Newton steps as the regularization is always "on" (i.e. $\sigma_k > 0$, $\forall k \geq 0$).

[2]More sophisticated update schemes might well be able to untangle the two effects but this is not subject of this thesis

3.2 Finding the Cubically Regularized Newton Step

The following analysis is rather lengthy and detailed but at the same time important as the subproblem minimization task commonly causes the main computation complexity of trust region/cubic regularization methods and is furthermore decisive for their convergence results. Our approach is mainly based on ideas developed for trust regions algorithms as in [Conn et al., 2000] Section 7.3 and [Nocedal and Wright, 2006] Section 4.3. Furthermore, we here aim to extend [Cartis et al., 2011] Section 6 for the sake of comprehensibility, since surprisingly many aspects of the exact ARC subproblem minimization are not given by Cartis et al. [2011] (including the final algorithm) but differences to the trust region approach are ubiquitous due to a) the absence of interior solutions and b) the difference in the definition of λ.

Readers who are short on time may view the exact subproblem minimization (Section 3.2.2) as an iterative **black box** routine that requires factorizing $B + \lambda I$ in each of its sub-iteration i, whose number grows in the desired accuracy[3]. Thus[4], it returns the global minimizer s^* in $O(id^3)$ operations for general matrices B and in $O(id)$ operations for tri-diagonal matrices B which, as we shall see, happens to be the case in Krylov subspaces.

3.2.1 On the Existence of a Global Minimizer

We shall start this Section by investigating the feasibility of the subproblems that arise in each SCR iteration.

Theorem 3.1 (Feasibility of the subproblem) *Let $f \in C^2$ and m_k defined as in (3.7). The problem*

$$\min_{s \in \mathbb{R}^d} m_k(s) \tag{3.7}$$

is feasible.

Proof: The model objective itself is a continuous function since (i) the dot product $s^\intercal g$ is a linear functional and bounded by the Cauchy-Schwarz inequality and (ii) $s^\intercal H_k s$ is continuous as long as the partial derivatives of f which define H are continuous functions which is given due to the assumption $f \in C^2$.

[3] At a globally linear and locally logarithmic rate (Theorem 6.3, [Cartis et al., 2011])

[4] *unless* B is indefinite and g orthogonal to the leftmost eigenvectors of B

In addition, m_k is coercive in the sense that for every sequence $\{s^\nu\} \in \mathbb{R}^d$ with $\|s^\nu\| \to \infty$ also $m_k(s^\nu) \to \infty$ since:

$$\begin{aligned} m_k(s^\nu) &= f(w_k) + (s^\nu)^\intercal g_k + \frac{1}{2}(s^\nu)^\intercal B_k s^\nu + \frac{\sigma_k}{3}\|s^\nu\|^3 \\ &\geq -\|s^\nu\|\|g_k\| + \frac{1}{2}\lambda_1\|s^\nu\|^2 + \frac{\sigma_k}{3}\|s^\nu\|^3 \\ &= \|s^\nu\|(-\|g_k\| + \|s^\nu\|(\lambda_{\min} + \frac{\sigma_k}{3}\|s^\nu\|)) \overset{\|s^\nu\|\to\infty}{\to} +\infty, \end{aligned} \tag{3.8}$$

where we used the above mentioned Cauchy-Schwarz inequality to lower bound the gradient- and a spectral decomposition to lower bound the Hessian term and furthermore applied that $\|g_k\|$ and $\lambda_{\min}$ are constant as well as $\sigma_k > 0$.

As a result of the continuity and coercivity of m_k we argue that each level set $\text{lev}^\alpha_\leq(m_k, \mathbb{R}^d) = \{s \in \mathbb{R}^d | m_k(s) \leq \alpha\}$ is non-empty and compact (see Lemma A.2 in the Appendix) which guarantees the existence of a global minimizer of

$$\min_{s\in\mathbb{R}^d} m_k(s) \quad s.t.\ s \in \text{lev}^\alpha_\leq(m_k, \mathbb{R}^d)$$

according to Weierstrass extreme value theorem (Theorem A.1 in Appendix). Finally, any global minimizer of m_k on the α level set is obviously also a global minimizer of m_k on $\mathbb{R}^d$ which proves the assertion.

□

Now that we know of the existence of (at least one) global minimizer let us proceed by giving its characterization, as it can be found in [Cartis et al., 2011] Theorem 3.1 for the proof.

Theorem 3.2 (Characterization of global minimizer) *Any s_k^* is a global minimizer of $m_k(s)$ over $\mathbb{R}^d$ if and only if it satisfies the system of equations*

$$(B_k + \lambda^* I)s_k^* = -g_k,\ \lambda^* = \sigma_k\|s_k^*\|,\ \textit{and}\ (B_k + \lambda^* I) \succeq 0, \tag{3.9}$$

where $\lambda_k^* = \sigma_k\|s_k^*\|$ and $B_k + \lambda_k^* I$ is p.s.d.

A direct consequence of this theorem is that since B_k is a quadratic matrix it has full rank whenever $B_k \succ 0$ and thus the solution of (3.7) is *unique* whenever B_k is positive definite. For the sake of simplicity, we shall drop the iteration subscript k in the following subsection.

3.2.2 Exact Subproblem Minimization

We shall now derive the useful proposition that the above mentioned problem can be solved by finding the root of an *univariate* non-linear equation in the scalar λ.

Problem formulation in λ

First off, let us use the optimality conditions (3.9) to express s as a function of λ and define a search window in which λ^* must lie. Obviously, from the third condition we have that $\lambda^* > -\lambda_{\min}(B)$. From the second condition we can deduce the equality

$$\|s(\lambda)\| = \frac{\lambda}{\sigma} \Leftrightarrow \|s(\lambda)\|^2 = \frac{\lambda^2}{\sigma^2}. \tag{3.10}$$

In order to study these conditions and the existence of a global minimizer in more detail, we assume for the moment that B is diagonizable and that its eigenvalue decomposition is given by $B = Q^\intercal \Lambda Q$. Furthermore, let $\lambda_1, \lambda_2, ..., \lambda_d$ denote the smallest to largest eigenvalues of B. Then, $B^{-1} = Q^\intercal \Lambda^{-1} Q$ since Q is orthogonal and the first condition is written

$$s(\lambda) = -(B + \lambda I)^{-1} g \Leftrightarrow s(\lambda) = -Q^\intercal (\Lambda + \lambda I)^{-1} Q g. \tag{3.11}$$

Since $Q^\intercal Q = I$ we can write

$$\begin{aligned} \|s(\lambda)\|^2 =& (-Q^\intercal (\Lambda + \lambda I)^{-1} Q g)^\intercal (-Q^\intercal (\Lambda + \lambda I)^{-1} Q g) \\ =& g^\intercal Q^\intercal ((\Lambda + \lambda I)^{-1})^\intercal (\Lambda + \lambda I)^{-1} Q g \\ =& \|(\Lambda + \lambda I)^{-1} Q g\|^2 \\ =& \sum_i^d \frac{(q_i^\intercal g)^2}{(\lambda_1 + \lambda)^2}, \end{aligned} \tag{3.12}$$

where q_i is the $i-$th row of Q and since B is symmetric it is the transpose of the $i-$th eigenvector of B. Since x^2 is a monotonic transformation on the image of $\|s\|$, this reveals some interesting properties of $\|s(\lambda)\|$. First of all, it is a non-negative, monotonically decreasing function in λ on the interval $(-\lambda_1, \infty)$ since then $\lambda_i + \lambda > 0$, for all $i = 1, \cdots, d$.

In fact, we have

$$\lim_{\lambda \to \infty} \|s(\lambda)\| = 0. \tag{3.13}$$

Secondly, $\|s(\lambda)\|$ has poles at each $\lambda = -\lambda_i$, for which the dot-product $q_i^\intercal g \neq 0$, i.e.

$$\lim_{\lambda \to -\lambda_i} \|s(\lambda)\| = \infty, \forall i \in \{i = 1, \cdots, d | \exists q_i^\intercal g \neq 0\} \tag{3.14}$$

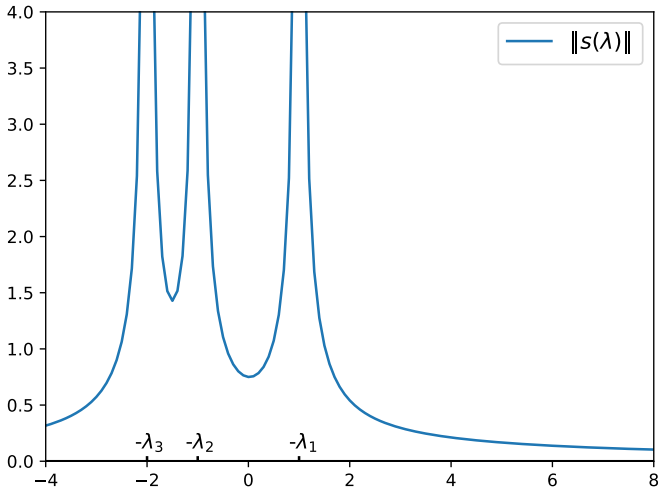


Figure 3.2: Graph of $\|s(\lambda)\|$ for $g = (1/2, 1/2, 1/2)^\intercal$ and $B = diag(2, 1, -1)$

Normal case

Lemma 3.3 *As long as* $\exists i \in \{i = 1, \dots, d | \lambda_i = \lambda_1\}$ *with* $q_i^\intercal g \neq 0$*,, there exists a unique solution*

$$\lambda^* \in [\max\{0, -\lambda_1\}, +\infty)], \tag{3.15}$$

which satisfies the global optimality conditions of (3.9).

Proof: Due to (3.14) and under the above mentioned assumptions $\|s(\lambda)\|$ has a pole at $\lambda = -\lambda_1$. Furthermore, we have already established that it is a monotonically decreasing function and the limit (3.13) holds. Thus, at some point λ^* to the right of $-\lambda_1$ (i.e. where $(B + \lambda^* I) \succ 0$) the stepsize norm is guaranteed to take the value λ^*/σ, i.e.

$$\|s(\lambda^*)\| = \lambda^*/\sigma \tag{3.16}$$

and the second and third condition of (3.9) are satisfied. Finally, in (3.11) we have defined $s(\lambda)$ in such a way that the first condition of (3.9) always holds. Thus, all of (3.9) is satisfied.

But we can narrow the search windows even further down. Note that whenever B is indefinite, $-\lambda_1 > 0$. Furthermore, when B is positive semidefinite $-\lambda_1 < 0$ but for $\lambda < 0$ we have $\lambda/\sigma < 0$ while $\|s(\lambda)\| \geq 0$ so there cannot be any negative root of θ_1. As a result, the value of λ we search for lies at the single positive root of $\theta_1(\lambda)$ in the interval.

□

Hard case When $q_i^T g = 0, \forall i \in \{i = 1, \ldots, d | \lambda_i = \lambda_1\}$, the limit (3.14) does not hold for $\lambda \to -\lambda_1$ and we cannot be sure that there actually is a $\lambda^* \in (-\lambda_1, \infty)$ such that $\|s(\lambda^*)\| = \lambda^*/\sigma$. As discussed above, for a positive semidefinite B the $\max\{0, -\lambda_1\} = 0$ and $\theta_1(0) \geq 0$ so that the root finding technique can be applied without further considerations.

However, for an indefinite B, we cannot be sure that $\|s(-\lambda_1)\| \geq -\lambda_1/\sigma$ and thus the function value $\theta_1(-\lambda_1)$ may be negative so that there exists no root in the given search interval. Inspired by Conn et al. [2000] we shall call this the hard case.

Lemma 3.4 *Be $q_i^T g = 0, \forall i \in \{i = 1, \ldots, d | \lambda_i = \lambda_1\}$ and B not positive definite. Then for any of these eigenvectors q_i, the step $s^* = s(-\lambda_1) + \alpha q_1$ with $\alpha \in \mathbb{R}$ chosen such that*

$$\lambda_1 = \sigma \|s(-\lambda_1) + \alpha q_1\|. \tag{3.17}$$

satisfies the global optimality conditions of (3.9).

Proof: Since $(B + \lambda I)$ must be positive semidefinite at any global minimizer s^*, we are only left with the choice $\lambda^* = -\lambda_1$. But then, the element in the sum term of equation (3.12) is undetermined and it cannot be used to find the minimizer s^*. However, at this point $(B + \lambda^* I)$ is positive semidefinite and singular. Thus, luckily, there are other solutions to (3.11), because for any eigenvector q_1 corresponding to the leftmost eigenvalue λ_1, we have $(B + \lambda^* I) q_1 = 0$ and thus

$$(B - \lambda_1 I)(s(-\lambda_1) + \alpha q_1) = -g \tag{3.18}$$

for any scalar α. Consequently, a model minimizer is given by (3.18) when $\alpha \in \mathbb{R}$ is chosen such that

$$\lambda_1 = \sigma \|s(-\lambda_1) + \alpha q_1\|, \tag{3.19}$$

since then $s^* = s(-\lambda_1) + \alpha q_1$ also satisfies (3.10).

□

As a consequence, resolving the hard case requires computing a partial eigensolution of B in order to obtain $-\lambda_1$ and solving a quadratic system of equations (3.19). However, note that the occurrence of a hard case is very unlikely, as it requires B to be indefinite and g to be orthogonal to the eigenvector(s) of the leftmost eigenvalue of B. Furthermore, it can be recognized without computing the eigenvalue decomposition of B as we shall elaborate when deriving Algorithm 3.

Computing λ^* with Newton's root finding algorithm

As derived above, we are looking for a value of $\lambda \geq \max\{0, -\lambda_1\}$ that solves

$$\theta_1(\lambda) := \|s(\lambda)\| - \frac{\lambda}{\sigma} = \sqrt{\sum_{i=1}^{d} \frac{(q_i^\intercal g)^2}{(\lambda_1 + \lambda)^2}} - \frac{\lambda}{\sigma} \stackrel{!}{=} 0 \tag{3.20}$$

Yet, for λ greater, but close to $-\lambda_1$, these functions are highly nonlinear as $\|s(\lambda)\|$ has a pole at $-\lambda_1$. Since Newton's method benefits from reasonably behaved derivatives in the area of interest, it produces more reliable and faster results when applied to the following reformulation of θ_1 as secular equations:

$$\varphi_1(\lambda) := \frac{1}{\|s(\lambda)\|} - \frac{\sigma}{\lambda} \stackrel{!}{=} 0, \tag{3.21}$$

since for λ slightly greater than $-\lambda_1$ this are nearly linear[5] and obviously the roots are the same. To see this compare $\theta_1(\lambda)$ and $\varphi_1(\lambda)$ in the neighborhood of $-\lambda_1$

$$\theta_1(\lambda) \approx \frac{c_1}{\lambda + \lambda_1} + c_2, \text{ and } \varphi_1(\lambda) \approx \frac{\lambda + \lambda_1}{c_3} + c_4 \tag{3.22}$$

for some constants $c_1, c_3 > 0$ and $c_2, c_4 \in \mathbb{R}$.

Figure 3.3 illustrates the graphs of the above discussed function for $\sigma = 1$, the indefinite Hessian $H = \operatorname{diag}(-1, 1)$ and the gradient $g = (1/4, 1)^\intercal$ on the left (soft case) as well as the same σ and H with the slightly altered gradient $g = (0, 1)^\intercal$. Note how the pole at $\lambda = 1$ vanishes and the intersect of $\|s(\lambda)\|$ and λ/σ moves to the left of $-\lambda_1$, which makes the root finding algorithm impractical.

As a result, for the soft case we are tempted to apply a plain root-finding Newton's method to the function variant φ_1 which generates a sequence of iterates λ^l by setting

$$\lambda^{l+1} = \lambda^l - \frac{\varphi_1(\lambda^l)}{\varphi_1'(\lambda^l)}. \tag{3.23}$$

Clearly, the evaluation of $\varphi_1(\lambda^l)$ involves the quantity $s(\lambda^l)$ and thus the solution of the system of equations[6]

$$(B + \lambda^l I)s(\lambda^l) = -g \tag{3.24}$$

[5] Wherever $\|s(\lambda)\|$ has a pole, $1/\|s(\lambda)\|$ has a root

[6] The eigendecomposition of (3.11) was just used to study the existence of a global solution. Applying it in the root finding steps would be too costly ($O(p^3 + p^2 \log^2 p \log b)$ for an approximation within 2^{-b} Pan and Chen [1999]

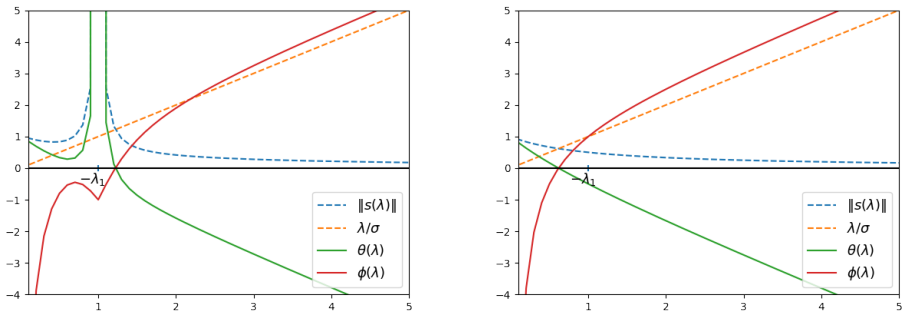


Figure 3.3: Graphs of the functions $\theta_1(\lambda)$, $\|s(\lambda)\|, \lambda/\sigma$ and $\varphi_1(\lambda)$ for the soft- and hard case respectively.

in each iteration l. If we had the inverse of $(B + \lambda^l I)$, which is positive definite in the range $\lambda > -\lambda_1$, the solution would be easily computable by a single matrix-vector product ($2d^2$ operations) . However, to get the inverse of this matrix, e.g. by Jordan-Gauss elimination, $O(2d^3/3)$ operations are needed. Alternatively, the Cholesky decomposition $(B + \lambda^l I) = L(\lambda^l)L(\lambda^l)^\intercal$ can be obtained in roughly half the number of operations ($O(d^3/3)$ operations and we can reformulate (3.24) into

$$L(\lambda^l)L(\lambda^l)^\intercal s(\lambda^l) = -g. \tag{3.25}$$

Because L is lower-triangular, we can rewrite this as two triangular systems which are solvable in d^2 operations each:

$$L(\lambda^l)y = -g \text{ (forward substitution) and } L(\lambda^l)^\intercal s(\lambda^l) = y \text{ (backward substitution).} \tag{3.26}$$

See Chapter 3 in Golub and Van Loan [2012] for more details on the complexity of solving general linear systems. Additionally, the Newton iteration requires the derivative of $\varphi_1(\lambda)$ with respect to λ but this can be obtained easily after factorizing $(B + \lambda^l I)$, as we shall see in the following.

Lemma 3.5 *Suppose that $s(\lambda)$ satisfies (3.11). Furthermore be $g \neq 0$. Then the function $\varphi_1(\lambda)$ is strictly increasing and concave when $\lambda > \max\{0, -\lambda_1\}$. Its first derivative is*

$$\varphi_1(\sigma)' = -\frac{s(\lambda)^\intercal \nabla_\lambda s(\lambda)}{\|s(\lambda)\|^3} + \frac{\sigma}{\lambda^2}, \tag{3.27}$$

where

$$\nabla_\lambda s(\lambda) = -(B + \lambda I)^{-1} s(\lambda). \tag{3.28}$$

Proof: First, from (3.12) we know that $\|s(\lambda)\|$ is strictly positive and decreasing in λ for $g \neq 0$ and $\lambda > -\lambda$. Furthermore $-\frac{\sigma}{\lambda}$ is strictly increasing

for $\lambda \neq 0$. Together, this implies that $\varphi_1(\sigma)$ is strictly increasing, when $\lambda > \max\{0, -\lambda_1\}$. The concaveness of $\varphi_1(\sigma)$ can be proven by showing that its second derivative is always lower or equal to zero (see Lemma 7.3.1 in Conn et al. [2000]).

Second, we have

$$\frac{\partial(\|s(\lambda)\|^2)}{\partial\lambda} = \frac{\partial(s(\lambda)^\intercal s(\lambda))}{\partial\lambda} = 2s(\lambda)^\intercal \nabla_\lambda s(\lambda) \tag{3.29}$$

and

$$\|s(\lambda)\|^{-1} = (\|s(\lambda)\|^2)^{-1/2}. \tag{3.30}$$

Thus, we can find the derivative of $\varphi_1(\sigma)$ by the chain rule

$$\varphi_1(\lambda)' = -\frac{1}{2}(\|s(\lambda)\|^2)^{-3/2} \cdot 2s(\lambda)^\intercal \nabla_\lambda s(\lambda) + \sigma\lambda^{-2}, \tag{3.31}$$

which gives (3.27). Finally, differentiating the defining equation

$$(B + \lambda I)s(\lambda) = -g \tag{3.32}$$

with respect to λ, gives

$$\begin{aligned} &\nabla_\lambda s(\lambda)(B + \lambda I) + s(\lambda)I = 0 \\ \Leftrightarrow\ &\nabla_\lambda s(\lambda) = -(B + \lambda I)^{-1} s(\lambda), \end{aligned} \tag{3.33}$$

which is equation (3.28).

□

As a matter of fact, we do not even need to find $\nabla_\lambda s(\lambda)$, because the numerator of the first summand of $\varphi_1(\lambda)'$ can be obtained in the following way

$$\begin{aligned} s(\lambda)^\intercal \nabla_\lambda s(\lambda) &= -s(\lambda)^\intercal [(L(\lambda)^\intercal)^{-1} L(\lambda)^{-1} s(\lambda)] \\ &= -[L(\lambda)^{-1} s(\lambda)]^\intercal [L(\lambda)^{-1} s(\lambda)] =: -\|w\|^2, \end{aligned} \tag{3.34}$$

which motivates step 5 of Newton's root finding algorithm as we would like to apply it to solve $\varphi(\lambda)$

Alternative update rule for the root finder

As argued above we solve $\varphi(\lambda) = 1/\|s\| - \sigma/\lambda$ instead of $\theta(\lambda) = \|s\| - \lambda/\sigma$ because we suspect φ to be more linear around λ^*. This idea originates from the trust region framework, where $\varphi(\lambda) = 1/\|s\| - \Delta$. However, for cubically regularized methods the last term of φ is no longer a constant but the ratio λ/σ. Thus, when $\lambda \to 0$, $\varphi(\lambda) \to -\infty$ and becomes very steep

Algorithm 2 Newton's method to solve $\varphi_1(\lambda) = 0$

Let $\lambda > \max\{0, -\lambda_1\}$ and $\lambda^0, \sigma > 0$ given.
for $l = 0, 1, 2, \ldots$ **do**:
    Factorize $(B + \lambda^l I) = L(\lambda^l)L(\lambda^l)^\intercal$
    Solve $L(\lambda^l)L(\lambda^l)^\intercal s(\lambda^l) = -g$
    Solve $L(\lambda^l)w = s(\lambda^l)$
    Set

$$\lambda^{l+1} = \lambda^l - \frac{\varphi_1(\lambda^l)}{\varphi_1'(\lambda^l)} = \lambda^l - \frac{\|s(\lambda^l)\|^{-1} - \sigma/\lambda^l}{\|w\|^2\|s(\lambda^l)\|^{-3} + \sigma/(\lambda^l)^2}. \tag{3.35}$$

and non-linear. This causes the Newton's root finding algorithm to generally take more steps in this setting than in the trust region case, especially when H is positive definite. Because then the poles of θ are in the negative region and λ^* may lie close to 0. Figure 3.4 illustrates this issue. In both frameworks we try to find the intersect between the dashed-purple curve with the dashed-green (ARC) and dashed-yellow (TR) curve by finding the root of the solid-cyan (ARC) and solid-blue (TR) curve. It becomes evident that the first derivative of φ is much more well behaved in the trust region case.

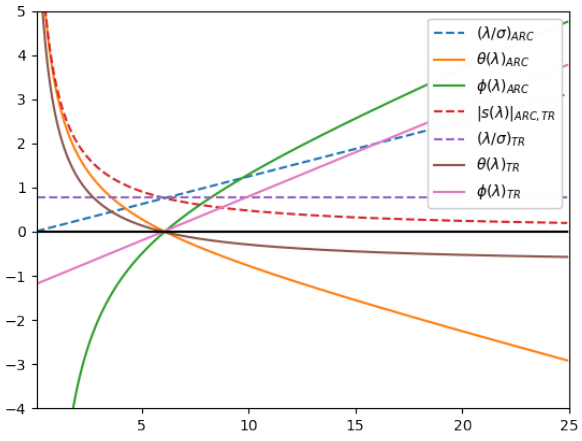


Figure 3.4: Functions involved in the subproblem minimization process of ARC and TR for $g = (1/4, 1/2)^\intercal$ and $H = diag(2, 1/2)$

It is thus meaningful to only linearize the first term $1/\sqrt{\varphi(\lambda)} = 1/\|s(\lambda)\|$ of $\varphi(\lambda)$ in order to reduce the number of Newton steps. The update $(\Delta\lambda)$

then needs to satisfy the following quadratic equation (compare [Cartis et al., 2011] Eq. (6.12))

$$-\frac{1}{2}\frac{\varphi'(\lambda^l)}{\varphi(\lambda^l)^3}(\Delta\lambda)^2+\left(\frac{1}{\varphi(\lambda^l)}-\frac{1}{2}\frac{\varphi'(\lambda^l)\lambda^l}{\varphi(\lambda^l)^3}\right)(\Delta\lambda)+\frac{\lambda^l}{\varphi(\lambda^l)}-\sigma=0, \quad (3.36)$$

which has also empirically proven to perform better than updating λ^l as in 3.35.

Safeguards

Unfortunately, Algorithm 2is only guaranteed to converge when started in a certain range of the search window $[0,-\lambda_1),+\infty)]$ and we thus must establish safeguards to guarantee its convergence globally. To this end, let us follow the approach of Conn et al. [2000] and separate the possible values of λ into three disjoint sets

$$\begin{aligned}\mathcal{N}&=\{\lambda \mid \lambda \leq \max\{0,-\lambda_1\}\},\\ \mathcal{L}&=\{\lambda \mid \max\{0,-\lambda_1\}<\lambda\leq\lambda^*\},\\ \mathcal{G}&=\{\lambda \mid \lambda>\lambda^*\}.\end{aligned} \quad (3.37)$$

These sets can be distinguished in the following way. Obviously, whenever $\lambda<0$ it lies in $\mathcal{N}$. For a positive λ, the Cholesky factorization only succeeds if $\lambda>-\lambda_1$. Thus, if it does not succeed we are in $\mathcal{N}$, else we are in $\mathcal{L}$ if $\varphi(\lambda)<0$ and in $\mathcal{G}$ if the function value is strictly positive. Note that, in the hard case there may not even be any $\lambda>-\lambda_1$ for which $\varphi(\lambda)<0$ and thus $\mathcal{L}=\emptyset$. This observation will be crucial for identifying the hard case algorithmically.

The convergence of Algorithm 2 can be guaranteed for the case that $\lambda_0\in\mathcal{L}$, since in this region the slope of φ is positive and hence the next iterate lies to the right of the prior ($\lambda^l<\lambda^{l+1}$). Furthermore, we start to the left of λ^* and cannot overshoot because of the concavity of φ (see Lemma 7.3.2 in Conn et al. [2000]). However, if $\varphi(\lambda^0)>0$, i.e. $\lambda^0\in\mathcal{G}$ the next iterate λ^+ will move left and furthermore have a negative function value because of the concavity of φ. That is, λ^+ may either lie in the just presented convergence regime or else $\lambda^l<-\lambda_1$. In the latter case the Cholesky factorization will not succeed and the convergence to λ^* is lost. Thus, inspired by [Conn et al., 2000] Section 7.3.4. we will find and update an interval $[\lambda_l,\lambda_u]$ in which λ^* is guaranteed to lie and reject any iterate λ^l that lies outside of this interval. Particularly, we proceed in the following way.

How to update λ? If $\lambda\in\mathcal{L}\cup\mathcal{G}$ we make a Newton step by solving 3.36 and accept it, if it lands in $\mathcal{L}$. Otherwise it must have landed in $\mathcal{N}$ and we instead guess a random λ in $[\lambda_u,\lambda_l]$.

How to update interval? Updating the interval in each iteration l is as easy as setting $\lambda_u := \lambda^l$, if $\lambda^l \in \mathcal{G}$ (where we landed by guess) and $\lambda_l := \lambda^l$ in any other case. This ensures that the interval shrinks in each iteration and in the worst case these bounds ultimately coincide. Specifically, if no iterate $\lambda_l > -\lambda_1$ gives a positive function value $\varphi(\lambda^l) > 0$ the safeguard interval is decreased repeatedly until it finally only contains $-\lambda_1$. Then, if $\varphi(-\lambda_1) < 0$ we know that we are in the hard case, else $\lambda^* = -\lambda_1$.

How to guess λ? In cases, where the Newton step is rejected, we shall update λ in analogy to Conn et al. [2000] Section 7.3.6

$$\lambda^{l+1} = \max\left\{\sqrt{\lambda_L \lambda_U}, \lambda_L + \theta(\lambda_U - \lambda_L)\right\}, \tag{3.38}$$

for some small $\theta \in (0,1)$, e.g. $\theta = 0.01$. This rule guarantees that the ratio of lengths of two successive intervals is

$$\max\{1-\theta, \theta, \frac{\sqrt{\lambda_u}}{\sqrt{\lambda_u}+\sqrt{\lambda_l}}\} \tag{3.39}$$

and it is biased towards finding $\lambda^{l+1} \in \mathcal{L}$.

How to initialize λ_l, λ_u and λ? As established above, the value λ^* we search must lie in the interval $[\max\{0, -\lambda_1\}, +\infty]$ which gives rise to a fairly large search windows. Fortunately, we can narrow it down significantly by following the approach for trust region methods as presented in [Conn et al., 2000]. However, *contrary* to the trust region case, we do not have the simple bound $\|s\| \leq \Delta$ which is why we will have to put some more effort into finding good initial bounds for λ^*, as we shall see now.

Obviously, zero as well as any lower bound on the negative eigenvalue of B serves as an initial value for λ_l. Since $\lambda_1 < \lambda_2 < \cdots < \lambda_d$, a negated upper bound on the largest eigenvalue would give such a lower bound on λ_1. However, for any non-negative definite matrix this bound will be positive and hence the negated form will be below 0 and thus of no use. A more promising approach is combining the Rayleigh inequalities of B and λI (see Definition A.9 in the Appendix)

$$\begin{aligned} \lambda_1 &\leq \frac{s^\intercal B s}{s^\intercal s} \leq \lambda_d \\ \lambda^* &\leq \frac{s^\intercal (\lambda^* I) s}{s^\intercal s} \leq \lambda^*, \end{aligned} \tag{3.40}$$

which sum up to

$$\lambda^* + \lambda_1 \leq \frac{s^\intercal (B + \lambda^* I) s}{s^\intercal s} \leq \lambda^* + \lambda_d. \tag{3.41}$$

Let us square the whole equation and recall the optimality characterization (3.9) which gives

$$\begin{aligned}(\lambda^* + \lambda_1)^2 \leq \frac{((B+\lambda^* I)s)^\intercal((B+\lambda^* I)s)}{s^\intercal s} \leq (\lambda^* + \lambda_d)^2 \\ \Leftrightarrow\ (\lambda^* + \lambda_1) \leq \frac{\|g\|}{\|s\|} \leq (\lambda^* + \lambda_d) \\ \Leftrightarrow\ (\lambda^* + \lambda_1) \leq \frac{\|g\|}{\lambda/\sigma} \leq (\lambda^* + \lambda_d).\end{aligned} \tag{3.42}$$

From which we can deduce the following two quadratic inequalities

$$\begin{aligned}(\lambda^*)^2 + \lambda_1\lambda^* - \sigma\|g\| \leq 0 \\ (\lambda^*)^2 + \lambda_d\lambda^* - \sigma\|g\| \geq 0,\end{aligned} \tag{3.43}$$

that are also valid for any lower bound on λ_1 and upper bound on λ_d. The first inequality holds for values of λ that satisfy

$$\frac{-\lambda_1 - \sqrt{\lambda_1^2 + 4\sigma\|g\|}}{2} \leq \lambda \leq \frac{-\lambda_1 + \sqrt{\lambda_1^2 + 4\sigma\|g\|}}{2} \tag{3.44}$$

and the second inequality holds for

$$\lambda \leq \frac{-\lambda_n - \sqrt{\lambda_n^2 + 4\sigma\|g\|}}{2} \text{ or } \lambda \geq \frac{-\lambda_n + \sqrt{\lambda_n^2 + 4\sigma\|g\|}}{2}. \tag{3.45}$$

Note that the left-hand sides of both, (3.44) and (3.45), are always non-positive and thus of no use for our purpose. The right-hand sides, however, constitute easily computable non-negative bounds on λ^* that give rise to a reasonably sized search window.

Of course, obtaining the exact values λ_1 and λ_d would require a full eigendecomposition of B and make the whole approach obsolete but as stated above it is perfectly fine to use the following bounds given by the Gershgorin intervals for real-symmetric matrices (Chapter 2,[Conn et al., 2000])

$$\begin{aligned}\lambda_1 \geq \min_{i=1,\dots,p}\left(B_{i,i} - \sum_{j=1, j\neq i}^{p} |B_{i,i}|\right) =: \Lambda_l \\ \lambda_n \leq \max_{i=1,\dots,p}\left(B_{i,i} + \sum_{j=1, j\neq i}^{p} |B_{i,i}|\right) =: \Lambda_u.\end{aligned} \tag{3.46}$$

Finally, let us note that a necessary condition for $B + \lambda I \succeq 0$ is that any diagonal entry of it be positive, which gives the extra condition $\lambda_l \geq -\min_i\{B_{i,i}\}$.

Together, these considerations lead to the following choice of λ_l and λ_u:

$$\begin{aligned} \lambda_l &= \max\left[-\min_i B_{i,i}, \frac{-\Lambda_u + \sqrt{\Lambda_u^2 + 4\sigma\|g\|}}{2}\right] \\ \lambda_u &= \frac{-\Lambda_l + \sqrt{\Lambda_l^2 + 4\sigma\|g\|}}{2} \end{aligned} \tag{3.47}$$

Of course, it is meaningful to replace Λ_u by upper bounds on the ℓ_2 norm of B like the Forbenius or ℓ_∞ norm in case they give sharper bounds on λ_d. Regarding the initial value of λ itself, without further information, we can in general not do better than guess any λ within the specified interval. However, in the case of an unsuccessful iterations, where the model has not changed but only the penalty parameter has risen, the terminating value for λ for the smaller σ should be chosen for both λ_l and the initial λ for the increased σ. This is because (in the easy case) this terminating λ will lie in $\mathcal{L}$ for the new (increased) penalty parameter, and convergence is guaranteed from here[7]. In the hard case, λ_l still gives a good lower bound.

The complete algorithm

We are now ready to state a complete algorithm to solve (3.7) and give its convergence guarantee. The algorithm is inspired by Algorithm 7.3.4 in [Conn et al., 2000] and can be viewed as an adaption to the cubic regularization case. Note that the dominant cost of this algorithm lie in the (repeated) factorization of $B + \lambda I$ which can be done in $O(d^3)$ flops for general matrices B and $O(d)$ flops for tri-diagonal matrices T.

[7] Graphically, $\|s(\lambda)\|$ does not change but the line λ/σ becomes flatter so the new intersect lies to the *right* of λ^l. Furthermore, $-\lambda_1$ has not changed so λ^l must lie in $\mathcal{L}$

Algorithm 3 Safeguarded Newton's method to solve $\varphi_1(\lambda) = 0$ at w_k

Let B, g and $\sigma > 0$ be given and Initialize λ_l, λ_u as in (3.47).
if previous iteration successful **then**:
    Choose a λ_0 in $[\lambda_l, \lambda_u]$ at random.
else:
    Set λ_0 to the terminating value of λ of the previous iteration
for $l = 0, 1, 2, \ldots$ **do**:
    `lambda+_in_N`:= False
    Attempt to factorize $(B + \lambda^l I) = L(\lambda^l)L(\lambda^l)^\intercal$
    if Factorization succeeds **then**:
        Solve $L(\lambda^l)L(\lambda^l)^\intercal s(\lambda^l) = -g$
        Solve $L(\lambda^l)w = s(\lambda^l)$
        Compute $\varphi(\lambda^l) = 1/\|s\| - \sigma/\lambda^l$ and check for termination
        Obtain the update $\Delta\lambda$ by solving Eq. (3.36) and set

$$\lambda^+ = \lambda^l + \Delta\lambda \tag{3.48}$$

        if $\varphi(\lambda^l) < 0$ **then** (1. $\lambda^l \in \mathcal{L}$):
            Set $\lambda^{l+1} := \lambda^+$
        else if $\varphi(\lambda^l) > 0$ **then** (2. $\lambda^l \in \mathcal{G}$):
            Set $\lambda_u := \lambda^l$
            if $\lambda^+ > 0$ **then**:
                Attempt to Factorize $(B + \lambda^+ I) = L(\lambda^l)L(\lambda^l)^\intercal$
                if Factorization succeeds **then** (2.1. $\lambda^+ \in \mathcal{L}$):
                    Set $\lambda^{l+1} := \lambda^+$
                else:
                    Set `lambda+_in_N`:=True
            else if $\lambda^+ > 0$ or `lambda+_in_N`==True **then** (2.2 $\lambda^+ \in \mathcal{N}$):
                Set $\lambda_l := \max\{\lambda_l, \lambda^+\}$
                $\lambda^{l+1} = \max\{\sqrt{\lambda_l * \lambda_u}, \lambda_l + 0.01 * (\lambda_u - \lambda_l)\}$
                if $\lambda_l == \lambda_u$ **then** (hard case):
                    Obtain λ_1, u_1 via partial eigendecomposition of B
                    Find an α by solving the quadratic equation

$$\|s + \alpha u_1\| = \lambda/\sigma \tag{3.49}$$

                    Set $s := s + \alpha u_1$
                    break
    else (3. $\lambda^l \in \mathcal{N}$):
        Set $\lambda_l := \max\{\lambda_l, \lambda^l\}$
        $\lambda^{l+1} = \max\{\sqrt{\lambda_l * \lambda_u}, \lambda_l + 0.01 * (\lambda_u - \lambda_l)\}$
        if $\lambda_l == \lambda_u$ **then**(hard case):
            Obtain λ_1, u_1 via partial eigendecomposition of B
            if $\lambda_1 >=$ **then** (w_k is 2nd order critical):
                break
            else hard case :
                Find an α by solving Eq. (3.49)
                Set $s := s + \alpha u_1$
                break
return s

Theorem 3.6 *Suppose that no termination test is applied in Step 12 of Algorithm 3. Then the iterates λ^l converge to λ^* and the limiting point s^l is s^*. The algorithm converges either in a finite number of steps or, except in the hard case, ultimately at a Q-quadratic rate.*

Proof: As described at the beginning of Section 3.2.2, there are three possible scenarios. First, if an iterate λ^l falls into the set $\mathcal{L}$, Lemma 7.3.2. in Conn et al. [2000] shows that all further iterates stay in this set and converge asymptotically Q-quadratic to λ^*. Finally, as established above, the corresponding step $s(\lambda^*)$ satisfies the optimality characterization (3.9).

Second, if λ^l lies in $\mathcal{G}$ the safety interval is reduced and the following iterate either falls into $\mathcal{L}$ or $\mathcal{N}$.

Third, if $\lambda^l \in \mathcal{N}$ the next iterate λ^{l+1} will be guessed within the safety interval according to (3.38). Unless $\lambda^{l+1} \in \mathcal{L}$, this will lead to either an increase in λ_u or a decrease in λ_l, depending on where λ^{l+1} falls, and thus gives a guaranteed reduction in the length of the safety interval as in (3.39). Hence, if no iterate falls into $\mathcal{L}$, the length of the interval converges to 0. Per design, $\lambda_l \leq -\lambda_1 \leq \lambda_u$ and thus the safety interval converges to $-\lambda_1$. This can only happen in the hard case. Since $H(-\lambda_1)$ is positive definite and singular, $(H - \lambda_1)u_1 = 0$ for any eigenvector u_1 corresponding to λ_1. Thus the condition $(H - \lambda_1)s = -g$ has many solutions, i.e.

$$(H + \lambda)(s + \alpha u_1) = -g \tag{3.50}$$

for any $\alpha \in \mathbb{R}$. Consequently, a model minimizer is given if we chose α such that

$$\|s + \alpha u_1\| = \lambda/\sigma \tag{3.51}$$

□

In conclusion, the exact subproblem solver is able to find the global minimizer s^* if run infinitely. Moreover, it converges at a quadratic rate once an iterate falls into $\mathcal{L}$ since only Newton steps are taken in the following subiterations. To make it practical, however, we break the routine in line 12 as soon as

$$|\varphi(\lambda^l)| \leq \varepsilon_{exact}, \tag{3.52}$$

for some $\varepsilon_{exact} > 0$.

3.2.3 Approximate Model Minimization

As described above the exact subproblem minimization routine, requires factorizing $B + \lambda I$ in each of its inner iterations, which makes the method prohibitively expensive for large scale learning as the cost rise cubically in the dimensionality of the problem. Thus it is an obvious alternative to minimize 3.7 only *approximately*.

In this regard, Cartis et al. [2011] show that it is indeed possible to retain the remarkable properties of Nesterov's cubic regularization algorithm as long as the inexact minimizer s_k satisfies the following two requirements

Assumption 3.7 (Approximate model minimizer)

$$s_k^\intercal g_k + s_k^\intercal B_k s_k + \sigma_k \|s_k\|^3 = 0 \tag{3.53}$$

$$s_k^\intercal B_k s_k + \sigma_k \|s_k\|^3 \geq 0 \tag{3.54}$$

which directly transfer to the SCR framework (originally (3.11) and (3.12) in [Cartis et al., 2011]). The first equation is equal to $\nabla m_k(s_k)^\intercal s_k$ and the second to $s_k^\intercal \nabla^2 m_k(s_k) s_k$. Thus they can be interpreted as variants of the first- and second-order criticality conditions when s_k is a global minimizer of m_k over a *subspace* of $\mathbb{R}^d$, which suggests the idea of minimizing (3.7) in a suitable space $\mathcal{L} \subseteq \mathbb{R}^d$.

One (extreme) way to do this would be to compute the minimizer of m_k along the current negative gradient direction $-g_k$ and step to the resulting point which is often referred to as the Cauchy step

$$s_k^C = -\alpha_k g_k, \text{ where } \alpha_k = \underset{\alpha \in \mathbb{R}_+}{\arg\min}\, m_k(-\alpha g_k). \tag{3.55}$$

While this can be done very cheaply, no second-order guarantees arise with this method[8] and the resulting first-order guarantees are no better than the ones of GD ([Cartis et al., 2011]). This is not particularly surprising since the Cauchy step does not make use of any curvature information. It is thus a natural idea to improve upon the Cauchy point and at the same time avoid the computational complexity of exact global minimization by minimizing in a subspace between the two extremes ($\mathbb{R}^1$ and $\mathbb{R}^d$).

Lemma 3.8 (Approximate model minimization) [9] *Let s_k be the global minimizer of $m_k(s)$, for $s \in \mathcal{L}_k$, where $\mathcal{L}_k \subseteq \mathbb{R}^d$. Then s_k satisfies Af3.7. Furthermore, letting Q_k denote any $n \times l$ matrix, whose columns form an orthonormal basis of $\mathcal{L}_k$, we have that*

$$Q_k^\intercal B_k Q_k + \sigma_k \|s_k\| I \succeq 0. \tag{3.56}$$

[8] Now is a good time to take another look at Figure 2.3 a)

[9] This is essentially Lemma 3.2 in [Cartis et al., 2011], but we here give a more detailed proof.

Proof: Since s_k is the global minimizer of m_k in $\mathcal{L}_k$ we can write

$$s_k = \arg\min_{s \in \mathcal{L}_k} m_k(s). \tag{3.57}$$

Furthermore we have

$$\text{col}(Q_k) = \text{span}(q_1, \ldots, q_l) = \mathcal{L}_k,\ q_i^\intercal q_j = 0, i \neq j \text{ and } \|q_i\| = 1,\ \forall i, j = 1, \ldots, l \tag{3.58}$$

where q_i is the $i-$th column of Q_k. As a result $Q_k^\intercal Q_k = I$ and Q_k is a regular matrix so that for all $s \in \mathcal{L}$ we can write

$$s = Q_k u,\ u \in \mathbb{R}^l. \tag{3.59}$$

Consequently, we can find the following subspace formulation of the model $m_k(u)$

$$u_k = \arg\min_{u \in \mathbb{R}^l} m_k(u) := f(w_k) + (Q_k u)^\intercal g_k + \frac{1}{2} u^\intercal Q_k^\intercal B_k Q_k u + \frac{1}{3}\sigma_k \|u\|^3, \tag{3.60}$$

where we used $\|Q_k u\|^2 = (Q_k u)^\intercal Q_k u = u^\intercal Q_k^\intercal Q_k u = u^\intercal u = \|u\|^2$ and thus

$$\|s\| = \|Q_k u\| = \|u\|, \text{ for all } u, s. \tag{3.61}$$

Taking the derivative of m_k with respect to u gives

$$Q_k^\intercal B_k Q_k u + \sigma_k \|u\| u = -Q_k^\intercal g_k \tag{3.62}$$

and multiplying by u yields

$$u^\intercal Q_k^\intercal B_k Q_k u + \sigma_k \|u\|^3 = -Q_k^\intercal g_k u, \tag{3.63}$$

which is equivalent to condition (3.53) in Assumption 3.7 due to (3.59). Furthermore, the second derivative of m_k with respect to u along with (3.61) and the fact that u_k is a global minimizer of m_k yields

$$Q_k^\intercal B_k Q_k + \sigma_k \|Q_k u_k\| I = Q_k^\intercal B_k Q_k + \sigma_k \|s_k\| I \succeq 0. \tag{3.64}$$

which proves (3.56). Finally, multiplying this equation by $u^\intercal u$ gives

$$u^\intercal Q_k^\intercal B_k Q_k u + \sigma_k \|s_k\|^3 \geq 0, \tag{3.65}$$

which is equivalent to condition (3.54) in Assumption 3.7 that is thus fulfilled by any global subspace minimizer $s_k = Q_k u_k$.

□

3.2.4 Krylov Subspace Minimization

Towards the end of finding a sufficiently accurate step s_k at reasonable cost Krylov-type methods are an attractive option as they naturally include the gradient in all iterations and allow to minimize m_k in increasingly larger subspaces and thus with increasing accuracy.

For trust region methods, this has first been proposed in Gould et al. [1999] as part of the so-called generalized Lanczos trust region method (GLTR), which takes a sequence of conjugate gradient steps until either the model minimizer is found (interior solution) or the size of the current trial step $s_{i,k}$ exceeds the trust region radius Δ_k. In the latter case it switches over to minimizing the $i-$dimensional Krylov subspace globally (boundary solution). The GLTR method is thus basically a hybrid between a conjugate gradient and a Lanczos-based approach. An important aspect to note is that both, the conjugate gradient and the Lanczos process build up a basis for the same nested Krylov spaces $\mathcal{K}_k = \{g_k, H_k g_k, H_k^2 g_k, \ldots\}$ and that it is possible to reconstruct the latter (orthogonal) from the former (conjugate) basis [Conn et al., 2000].

In the cubic regularization framework, however, interior solutions cannot arise and thus it is meaningful to directly use the Lanczos method to build up an orthogonal basis $Q_i = (q_1, q_2, \ldots, q_i)$ for the Krylov subspace $\mathcal{K}_k(B_k, g_k, i) = \{g_k, B_k g_k, B_k^2 g_k, ..., B_k^i g_k\}$, which can be done in a Hessian-free manner[10].

Algorithm 4 Lanczos method for an orthogonal basis of $\mathcal{K}(B, g, j)$

1: Given g_k, set $y_1 = g_k$ and $q_0 = 0$.
2: **for** $i = 1, 2, ...$ **do**:
3: $\quad \gamma_i = \|y_i\|$
4: $\quad q_i = y_i / \gamma_i$
5: $\quad \delta_i = q_i^\mathsf{T} B_k q_i$
6: $\quad y_{i+1} = B_k q_i - \delta_i q_i - \gamma_i q_{i-1}$

We note that, in matrix terms, the last equation can be written as follows

$$H Q_i - Q_i T_i = \gamma_{i+1} q_{i+1} e_{i+1}^\mathsf{T}, \tag{3.66}$$

where e_{i+1} is the $(i+1)-$th unit vector and the tridiagonal matrix T_i is

$$T_i = \begin{bmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \delta_{i-1} & \gamma_i \\ & & & \gamma_i & \delta_i \end{bmatrix}.$$

[10] For the sake of simplicity we drop the iteration subscript k for all quantities generated directly by the Lanczos process.

For the sake of brevity we refer the reader to Conn et al. [2000] Section 5.2 for more details on the derivation of Algorithm 4 and shall now explain how it can be used to approximately solve the cubic regularization subproblems. For each outer iteration k, successive problems of the type

$$s_{i,k} = \underset{s \in \mathcal{K}(g,H,i)}{\arg\min}\, m_k(s) \tag{3.67}$$

shall be solved exactly over increasing subspaces $\mathcal{K}_i$ until a sufficiently accurate solution is found.

Assumption 3.9 (Termination Criteria) *Assume that the Lanczos process stops as soon as some inner iteration i satisfies the criterion*

$$TC: \|\nabla m_k(s_{i,k})\| \leq \theta_k \|g_k\|, \tag{3.68}$$

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

Obviously, since $\nabla m_k^s(s_{i,k}) = 0$ for any global minimizer of m_k^s in $\mathbb{R}^p$, the worst that can happen is that the subproblem solver iterates until the global minimizer is found but in practice we hope that the subroutine terminates well before a Krylov space of dimensionality d is built up.

Krylov subproblems Another, crucial advantage of the Lanczos method over exact minimization is that a closer look at the subproblems (3.67) reveals that an equivalent formulation with a tridiagonal matrix can be found, which is a lot easier to factorize ($O(d)$) especially in high dimensions (see [Conn et al., 2000] 5.2.12). To show this we recall that the $i-$th iteration of the Lanczos process gives rise to an *orthogonal* matrix Q_i with $Q_i^\intercal Q_i = Q_i Q_i^\intercal = I$. Hence, Q_i is a regular matrix and for each vector $s \in \mathbb{R}^d$ exists a vector $u \in \mathbb{R}^i$ of the form $s = Q_i u$. Regarding these vectors we note the following key relationships:

$$(i)\ Q_i^\intercal g_k = \gamma_1 e_1,\ \ (ii)\ Q_i^\intercal B_k Q_i = T_i \text{ and } (iii)\ \|s\| = \|u\|. \tag{3.69}$$

(i) follows immediately from the definition of $q_1 = g/\|g\|$ and the fact that the q_i are orthonormal ($q_i^\intercal q_j = 0$, $i \neq j$). Premultiplying Eq. (3.66) by $Q_i^\intercal$ and using $Q_i^\intercal q_{i+1} = 0$ gives (ii). Finally, (iii) was already derived for Eq. (3.61).

We are thus able to reformulate (3.67) in the following way

$$u_i = \underset{u \in \mathbb{R}^i}{\arg\min}\, m_{i,k}(u) := f(w_k) + \gamma_1 u^\intercal e_1 + \frac{1}{2} u^\intercal T_i u + \frac{1}{3}\sigma_k \|u\|^3. \tag{3.70}$$

In each subiteration this $i-$dimensional problem is handed over to the exact minimization routine presented in the previous section (Algorithm 3) which can solve (3.70) in $O(l(i))$ flops (thanks to the tri-diagonal structure of T_i), where l is the number of inner solver iterations.

Of course, recovering $s_{k,i}$ from u_i involves the computation of the matrix-vector product $Q_i u_i$ and furthermore it may be prohibitively storage intensive to keep Q_i in memory for high dimensional problems. Fortunately, there is a simple way to test the stopping criterion (3.9) with readily-available iteration information. To this end, note that

$$\|\nabla_s m_k(s_{k,i})\| = \|g_k + Bs_{k,i} + \sigma_k \|s_{k,i}\| s_{k,i}\| = \|g_k + Bs_{k,i} + \lambda_{k,i} s_{k,i}\| \tag{3.71}$$

, where with Conn et al. [2000] Theorem 7.5.10 the right hand side can be shown to equal

$$\gamma_{k+1} |e_{k+1}^T u_k|. \tag{3.72}$$

We can thus rewrite the stopping criterion TC (3.9) such that the Lanczos process is terminated as soon as for some inner iterate i

$$\gamma_{i+1} |e_{i+1}^T u_i| \leq \kappa_\theta \min\{1, \|u_i\|\} \|g_k\| \tag{3.73}$$

and either reload Q from backing-storage or re-run the Lanczos process in order to recover $s_k = Q_i u_i$

3.3 Total Computational Complexity

As a result of the above analysis, each major SCR iteration k triggers i Lanczos process iterations (Algorithm 4) of which each triggers l_i exact subproblem iterations (Algorithm 3). The total number of operations that are due to the latter sum up to $O(\sum_{j=0}^{i} l_j(j))$. Obviously, $i < d$ and since the Newton root finder is linearly convergent (once $\lambda^l \in \mathcal{L}$) a rough bound on this would be $O(\log(1/\varepsilon_{exact})d)$. The operations that are needed to build up the i dimensional Lanczos process in the first place are $O(idn)$ due to the the Hessian-vector products $B_k q_i$ in Step 5. Finally, the outer SCR frameworks main effort lies in the computation of the function and model decrease which can be done in $O(nd)$.

Intuitively, we can expect i to be small in the beginning but grow asymptotically as more and more accurate steps are needed close to a minimizer. More precisely, the analysis in Chapter 4 will show that $i \to d$ asymptotically for second-order convergence. Thus, for high-dimensional problems where $d \gg 0$ and consequently $i \gg 0$ asymptotically as well as $i = d$ in the "worst case" the major computation cost of our method clearly lies in the Lanczos process which may require up to $O(d^2 n)$ operations[11]. Since these cost increase linearly in n we are thus confident to achieve substantial runtime reductions by sub-sampling the Hessian which reduces the (worst case) per-iteration cost to $O(d^2|\mathcal{S}_H|)$.

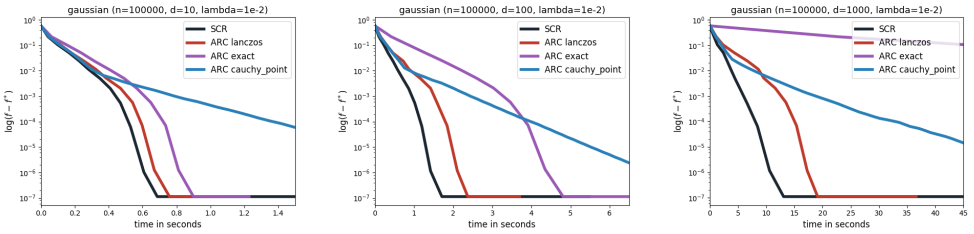
[11] Note that this is still significantly cheaper than a global exact minimization which requires $O(d^3)$ flops for factorizing B_k

For the case where $n \gg 1$ is so large that it becomes too expensive to even compute the full gradient ($O(nd)$) we will explicitly develop and include the possibility to use inexact gradient approximations in SCR. However, in "smaller" problems the above considerations suggest that it is indeed cost-effective to use the high-quality first-order information provided by the full gradients. This evolves from that fact that k gradient evaluations stand against a couple of hundred or even thousand Hessian-vector products and for this type of method k is likely to be very small (in the tens) especially compared to the number of gradient evaluations that first-order methods commonly take.[12] Interestingly, Martens [2010] who employ the conjugate-gradient method in each iteration of their damped Newton framework arrive at a similar conclusion and thus sub-sampled only Hessians as well.

Consider Figure 3.5 for an illustration of the above considerations. Here we minimize the convex regularized logistic empirical error on the synthetic dataset *gaussian* for which the feature vectors $X = (x_1, x_2, ..., x_d), x_i \in \mathbb{R}^n$ were drawn from a multivariate Gaussian distribution

$$X \sim \mathcal{N}(\mu, \Sigma) \tag{3.74}$$

with a mean of zero $\mu = (0, \ldots, 0)$ and the $n \times d$ identity matrix as covariance matrix, i.e. $\Sigma = I$. This gives rise to a very well-behaved problem [13]



$d = 10$ $d = 100$ $d = 1000$

Figure 3.5: Log suboptimality (avg. of 10 independent runs) on Gaussian datasets with $\Sigma = I$.

All methods were started from the initial weight vector $w_0 := (0, \ldots, 0)$ and take between 10 and 12 iterations to reach $\|\nabla f(w_k)\| < 10^{-11}$, except for the Cauchy step solver that takes much more iterations albeit the well-conditioning of the problem. Note how (i) the exact minimizer gives competitive results on the low dimensional dataset but suffers heavily from an

[12] where it is thus more likely beneficial to sub-sample the gradients

[13] algorithms face an average condition number of 14.7 on their way to- and 22.1 at the minimizer for $d = 1000$ and comparable numbers for the smaller problems)

increase in d and (ii) the runtime of the Lanczos solver can indeed be further reduced by sub-sampling the Hessian (SCR).

Chapter 4

Theoretical Analysis

In this section, we provide the convergence analysis of sub-sampled cubic regularization methods. For the sake of brevity, we assume global Lipschitz continuous Hessians immediately but note that a superlinear local convergence result (Theorem A.5) as well as the global first-order convergence theorem can both be obtained without the former assumption.

First, we lay out some basic Assumptions and present critical conditions regarding the exactness of general 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., 2011] but adapted for the case of inexact 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 4.1 (second-order continuity) *The functions $f_i \in C^2(\mathbb{R}^d)$, ∇f_i and $\nabla^2 f_i$ are Lipschitz continuous for all i, with Lipschitz constants κ_f, κ_g and κ_H respectively.*

By use of the triangle inequality, if follows that these assumptions hold for all $g = \frac{1}{|\mathcal{S}_g|}\sum_{i\in|\mathcal{S}_g|}\nabla f_i$ and $B = \frac{1}{|\mathcal{S}_B|}\sum_{i\in|\mathcal{S}_B|}\nabla^2 f_i$, independent of the actual samples $\mathcal{S}_g$ and $\mathcal{S}_H$. Furthermore, note that the Hessian and gradient norms are uniformly bounded as a consequence of Assumption 4.1, i.e. $\|\nabla f_i\| \leq \kappa_f$ and $\|\nabla^2 f_i\| \leq \kappa_g$ which of course translates to g and B.

4.2 Sampling Gradient and Hessian Information

We will now first consider conditions regarding the agreement of the stochastic gradient and stochastic Hessian with its deterministic counterparts that are *sufficient* for retaining the local and global convergence guarantees of ARC. Subsequently, we introduce probabilistic deviation bounds for these quantities based on which we shall translate the agreement conditions into concrete conditions on the magnitude of the samples in each iteration.

4.2.1 Sufficient Agreement Conditions

In each iteration k, the Hessian approximation B_k shall satisfy condition AM.4 from [Cartis et al., 2011], which we restate here for the sake of completeness.

Assumption 4.2 (Sufficient Agreement of H and B)

$$\|(B_k - H(w_k))s_k\| \leq C\|s_k\|^2, \forall k \geq 0, C > 0. \tag{4.1}$$

We explicitly stress the fact that this condition is stronger than the well-known Dennis Moré Condition[1], which usually characterizes superlinear convergence of quasi-Newton methods [Dennis and Moré, 1974]:

$$\frac{\|(B_k - H(w_k))\, s_k\|}{\|s_k\|} \to 0, \text{ whenever } \|g_k\| \to 0. \tag{4.2}$$

While quasi-Newton approximations satisfy the latter, there is no theoretical guarantee that they also satisfy the former [Cartis et al., 2011].

Similarly, any sub-sampled gradient shall satisfy the following condition which closely resembles the intuition of the Hessian agreement condition.

Assumption 4.3 (Sufficient Agreement of ∇f and g)

$$\|\nabla f(w_k) - g(w_k)\| \leq M\|s_k\|^2, \forall k \geq 0,\ M > 0. \tag{4.3}$$

4.2.2 Concentration Inequalities

In what follows we shall quickly introduce the idea of so-called concentration inequalities. The interested reader is referred to [Boucheron et al., 2013] for an extensive discourse. To put it briefly, these inequalities control the probability of a sum of general random variables to be far from its expectation. In a way they generalize classic limit theorems, such as the Laws of

[1] Intuitively, it states that B_k does not need to converge to $H(w)$ uniformly but that it suffices for B_k to become increasingly similar to the Hessian *along the directions* s_k.

Large Numbers[2] or the Central Limit Theorem, to a non-asymptotic setting which makes them specifically attractive for applications in machine learning. One of the simplest such inequalities is Chebyshev's inequality which follows directly from Markov's inequality:

Lemma 4.4 (Chebyshev's inequality) *Let X be a random variable with $\mathbb{E}\left[X^2\right] < \infty$. Then,*

$$P\left(|X - \mathbb{E}\left[X\right] > \varepsilon\right) \leq \frac{Var(X)}{\varepsilon^2} \tag{4.4}$$

The intuitive idea that arises form this inequality is that while a sum of *i.i.d.* centered random variables X_i

$$X = X_1 + X_2 + \ldots + X_n \tag{4.5}$$

may, if all X_i "work together", take values in the order of $O(n)$, it is actually very likely that some of the X_i cancel out such that X is just of order of its standard deviation $O(\sqrt{n})$.

More elaborated results such as Hoeffding's inequality and Bernstein's inequality (as well as the recent vector and Matrix versions) strengthen this result. In particular, they give (contrary to the *quadratically* decaying bound of 4.4) *exponentially* decaying bounds on the probability of a random variable to differ greater or equal than ε from its mean for any fixed number of samples. We here use Bernstein's inequality to upper bound the ℓ_2-norm distance $\|\nabla f - g\|$, as well as the spectral-norm distance $\|B - H\|$ by quantities involving the sample sizes $|\mathcal{S}_B|$ and $|\mathcal{S}_g|$. By applying the resulting bounds in the sufficient agreement assumptions (A4.2 & A4.3) and re-arranging for $|\mathcal{S}_B|$ and $|\mathcal{S}_g|$ respectively, we are able to translate these assumptions into concrete sampling conditions.

For the sake of simplicity we shall drop the iteration subscript k in this subsection.

Vector Bernstein Inequality First, we extend the Vector Bernstein Inequality as it can be found in Kueng and Gross [2014] to the *average* of independent, zero-mean vector-valued random variables. This result will be applied in Lemma 4.7 in order to find a probabilistic bound on the deviation of the sub-sampled gradient from the full gradient.

Lemma 4.5 (Vector Bernstein Inequality) *Let $x_1, \ldots, x_n$ be independent vector-valued random variables with common dimension d and assume that*

[2] which state that sums of independent random variables *concentrate* around their means

each one is centered, uniformly bounded and also the variance is bounded above:

$$\mathbb{E}[x_i] = 0 \text{ and } \|x_i\|_2 \leq \mu \text{ as well as } \mathbb{E}\left[\|x_i\|^2\right] \leq \sigma^2.$$

Let

$$z := \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Then we have for $0 < \varepsilon < \sigma^2/\mu$

$$P(\|z\| \geq \varepsilon) \leq \exp\left(-n \cdot \frac{\varepsilon^2}{8\sigma^2} + \frac{1}{4}\right) \tag{4.6}$$

Proof: Proposition 7 in Kueng and Gross [2014] gives the following Vector Bernstein inequality for values of $0 < \varepsilon < \sigma^2/\mu$.

$$P\left(\|\sum_{i=1}^{n} x_i\| \geq \varepsilon\right) \leq \exp\left(-\frac{\varepsilon^2}{8V} + \frac{1}{4}\right). \tag{4.7}$$

Since the individual variance is assumed to be bounded above, we can write

$$V = \sum \mathbb{E}\left[\|x_i\|^2\right] \leq n\sigma^2. \tag{4.8}$$

This term also constitutes an upper bound on the variance of $y = \sum_{i=1}^{n} x_i$, because the x_i are independent and thus uncorrelated . However, $z = \frac{1}{n}\sum_{i=1}^{n} x_i$ and we must account for the averaging term. Since the x_i are centered we have $\mathbb{E}[z] = 0$, and thus

$$\begin{aligned} Var(z) &= \mathbb{E}\left[\|z - \mathbb{E}[z]\|^2\right] = \mathbb{E}\left[\|z\|^2\right] = \mathbb{E}\left[\|\frac{1}{n}\sum_{i=1}^{n} x_i\|^2\right] = \frac{1}{n^2}\mathbb{E}\left[\sum_{i,j}\left(x_j^\intercal x_i\right)\right] \\ &= \frac{1}{n^2}\sum_{i,j}\left(\mathbb{E}\left[x_j^\intercal x_i\right]\right) = \frac{1}{n^2}\left(\sum_{i=1}^{n}(\mathbb{E}\left[x_i^\intercal x_i\right]) + \sum_{i=1}^{n}\sum_{j\neq i}^{n}(\mathbb{E}\left[x_i^\intercal x_j\right])\right) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}(\mathbb{E}\left[\|x_i\|^2\right]) \leq \frac{1}{n}\sigma^2, \end{aligned} \tag{4.9}$$

where we used the fact that the expectation of a sum equals the sum of the expectations and the crossterms $\mathbb{E}\left[x_j^\intercal x_i\right] = 0, j \neq i$ because of the independence assumption. Hence, we can bound the term

$$V \leq \frac{1}{n}\sigma^2 \tag{4.10}$$

for the random vector sum z.

Now, since $n > 1$ and $\varepsilon > 0$, as well as $P(z > \varepsilon)$ is falling in ε and $\exp(-\alpha)$ falling in $\alpha \in \mathbb{R}$, we can use the bound (4.10) on the variance of z in (4.7), which gives the desired inequality

$$P\left(\|z\| \geq \varepsilon\right) \leq \exp\left(-n \cdot \frac{\varepsilon^2}{8\sigma^2} + \frac{1}{4}\right) \tag{4.11}$$

□

Matrix Bernstein Inequality Similar to the vector case, the following result exhibits that sums of independent random matrices provide normal concentration near its mean in a range determined by the variance of the sum. In Lemma 4.9 we will apply this result in order to derive a bound on the deviation of the sub-sampled Hessian from the full Hessian.

Lemma 4.6 (Matrix Bernstein Inequality) *Let $A_1, .., A_n$ be independent random Hermitian matrices with common dimension $d \times d$ and assume that each one is centered, uniformly bounded and also the variance is bounded above:*

$$\mathbb{E}\left[A_i\right] = 0 \text{ and } \|A_i\|_2 \leq \mu \text{ as well as } \|\mathbb{E}\left[A_i^2\right]\|_2 \leq \sigma^2$$

Introduce the sum

$$Z := \frac{1}{n}\sum_{i=1}^{n} A_i$$

Then we have

$$P(\|Z\| \geq \varepsilon) \leq 2d \cdot \exp\left(-n \cdot \min\left\{\frac{\varepsilon^2}{4\sigma^2}, \frac{\varepsilon}{2\mu}\right\}\right) \tag{4.12}$$

Proof: Theorem 12 in Gross [2011] gives the following Operator-Bernstein inequality

$$P(\|\sum_{i=1}^{n} A_i\| \geq \varepsilon) \leq 2d \cdot \exp\left(-\min\left\{\frac{\varepsilon^2}{4V}, \frac{\varepsilon}{2\mu}\right\}\right), \tag{4.13}$$

where $V = n\sigma^2$. As we shall see, this is an upper bound on the variance of $Y = \sum_{i=1}^{n} A_i$ since the A_i are independent and have an expectation of zero

($\mathbb{E}[Y] = 0$).

$$\begin{aligned} Var(Y) =& \|\mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2\| = \|\mathbb{E}\left[(\sum_i A_i)^2\right]\| = \|\mathbb{E}\left[\sum_{i,j} A_i A_j\right]\| \\ =& \|\sum_{i,j}\mathbb{E}\left[A_i A_j\right]\| = \|\sum_i \mathbb{E}\left[A_i A_i\right] + \sum_i\sum_{j\neq i}\mathbb{E}\left[A_i A_j\right]\| \\ =& \|\sum_i \mathbb{E}\left[A_i^2\right]\| \leq \sum_i \|\mathbb{E}\left[A_i^2\right]\| \leq n\sigma^2, \end{aligned} \tag{4.14}$$

where we used the fact that the expectation of a sum equals the sum of the expectations and the crossterms $\mathbb{E}\left[A_j A_i\right] = 0, j \neq i$ because of the independence assumption.

However, $Z = \frac{1}{n}\sum_{i=1}^n A_i$ and we must account for the averaging term:

$$Var(Z) = \|\mathbb{E}\left[Z^2\right]\| = \|\mathbb{E}\left[(\frac{1}{n}\sum_{i=1}^n A_i)^2\right]\| = \frac{1}{n^2}\|\mathbb{E}\left[(\sum_{i=1}^n A_i)^2\right]\| \leq \frac{1}{n}\sigma^2. \tag{4.15}$$

Hence, we can bound

$$V \leq \frac{1}{n}\sigma^2 \tag{4.16}$$

for the *average* random matrix sum Z. Furthermore, since $n > 1$ and $\varepsilon, \mu > 0$ as well as $\exp(-\alpha)$ decreasing in $\alpha \in \mathbb{R}$ we have that

$$\exp\left(-\frac{\varepsilon}{2\mu}\right) \leq \exp\left(-\frac{\varepsilon}{n2\mu}\right). \tag{4.17}$$

Together with the Operator-Bernstein inequality, (4.16) and (4.17) give the desired inequality (4.12).

□

4.2.3 Sampling Conditions

Gradient Sampling

In each iteration k we shall sub-sample a gradient index set $\mathcal{S}_{g,k} \subseteq \{0, 1, \ldots, n\}$ uniformly, independently and without replacement. We then use the average of the gradients at these indices as an (unbiased) estimator of $\nabla f(w_k)$

$$g_k := \frac{1}{|\mathcal{S}_{g,k}|}\sum_{i\in\mathcal{S}_{g,k}} \nabla f_i(w_k). \tag{4.18}$$

Furthermore, let e_k be the gradient approximation error, i.e. $e_k := g_k - \nabla f(w_k)$. Before we derive the sampling condition let us quickly revisit some

basic properties of sub-sampled gradients which will come into play when analysing the convergence behaviour of methods with sub-sampled gradients. Specifically, note the following two properties regarding the gradient norm

$$\|\nabla f\|_2 = \sqrt{\nabla(f^1)^2 + ... + \nabla(f^d)^2}$$

, where $\nabla(f^j) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i^j$ is the average over all individual gradients in the $j-th$ dimension:

- neither $\|\nabla f\| = 0 \rightarrow \|g\| = 0$ nor $\|g\| = 0 \rightarrow \|\nabla f\| = 0$ hold
- neither $\|\nabla f\| \geq \varepsilon \rightarrow \|g\| \geq \varepsilon$ nor $\|g\| \geq \varepsilon \rightarrow \|\nabla f\| \geq \varepsilon$ hold.

However, regarding the deviation of the sub-sampled from the exact gradient we can state the following Lemma due to the vector Bernstein inequality.

Lemma 4.7 (Gradient deviation bound) *Let the sub-sampled gradient g_k be defined as in Eq. (4.18). Then we have with probability $(1-\delta)$ that*

$$\|g(w_k) - \nabla f(w_k)\| \leq 4\sqrt{2}\kappa_f \sqrt{\frac{\log((2d)/\delta) + 1/4}{|\mathcal{S}_{g,k}|}}. \tag{4.19}$$

Proof: To apply the vector Bernstein inequality (4.6) we need to center the gradients. Thus we define

$$x_i = g_i(w_k) - \nabla f(w_k),\ i = 1, \ldots, |\mathcal{S}_{g,k}| \tag{4.20}$$

and note that from the Lipschitz continuity of f (A4.1), we have

$$\|x_i\| = \|g_i(w_k) - \nabla f(w_k)\| \leq \|g_i(w_k)\| + \|\nabla f(w_k)\| \leq 2\kappa_f \tag{4.21}$$

as well as

$$\|x_i\|^2 \leq 4\kappa_f^2,\ i = 1, \ldots, |\mathcal{S}_{g,k}|. \tag{4.22}$$

With $\sigma^2 := 4\kappa_f^2$ and

$$z = \frac{1}{|\mathcal{S}_{g,k}|}\sum_{i\in\mathcal{S}_{g,k}} x_i = \frac{1}{|\mathcal{S}_{g,k}|}\sum_{i\in\mathcal{S}_{g,k}} g_i(w_k) - \frac{1}{|\mathcal{S}_{g,k}|}\sum_{i\in\mathcal{S}_{g,k}} \nabla f(w_k) = g(w_k) - \nabla f(w_k) \tag{4.23}$$

in equation (4.6), we can require the probability of a deviation larger or equal to ε to be lower than some $\delta \in (0, 1]$

$$\begin{aligned} P\left(\|g(w_k) - \nabla f(w_k)\| > \varepsilon\right) \leq & 2d \exp\left(-|\mathcal{S}_{g,k}| \cdot \frac{\varepsilon^2}{32\kappa_f^2} + \frac{1}{4}\right) \overset{!}{\leq} \delta \\ \Leftrightarrow & |\mathcal{S}_{g,k}| \cdot \frac{\varepsilon^2}{32\kappa_f^2} - \frac{1}{4} \overset{!}{\geq} \log((2d)/\delta) \\ \Leftrightarrow & \varepsilon \geq 4\sqrt{2}\kappa_f \sqrt{\frac{\log\left((2d)/\delta\right) + 1/4}{|\mathcal{S}_{g,k}|}}. \end{aligned} \tag{4.24}$$

Conversely, the probability of a deviation of

$$\varepsilon \leq 4\sqrt{2}\kappa_f \sqrt{\frac{\log\left((2d)/\delta\right) + 1/4}{|\mathcal{S}_{g,k}|}} \tag{4.25}$$

is lower or equal to $1 - \delta$.

□

This result 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. Of course, any sampling scheme that guarantees the right hand side of (4.19) to be smaller or equal to M times the squared step size, directly satisfies the sufficient gradient agreement condition (A4.3). Consequently, plugging the former into the latter and rearranging for the sample size gives the following Theorem.

Theorem 4.8 (Gradient Sampling) *If*

$$|\mathcal{S}_{g,k}| \geq \frac{32\kappa_f^2 \left(\log\left((2d)/\delta\right) + 1/4\right)}{M^2 \|s_k\|^4},\ M \geq 0 \text{ and } \forall k \geq 0 \tag{4.26}$$

then g_k satisfies the sufficient agreement condition A4.3 with probability $(1 - \delta)$.

Proof:

By use of Lemma 4.7 we can write

$$\begin{aligned} &\|g(w_k) - \nabla f(w_k)\| \leq M\|s\|^2 \\ \overset{w.h.p.}{\Leftrightarrow}\ &4\sqrt{2}\kappa_f \sqrt{\frac{\log((2d)/\delta + 1/4)}{|S_{g,k}|}} \leq M\|s_k\|^2 \\ &|\mathcal{S}_{g,k}| \geq \frac{32\kappa_f^2 \log\left((2d)/\delta + 1/4\right)}{M^2 \|s_k\|^4} \end{aligned} \tag{4.27}$$

□

Hessian Sampling

In each iteration k we shall sub-sample a Hessian index set $\mathcal{S}_{B,k} \subseteq \{0, 1, \ldots, n\}$ uniformly, independently and without replacement. We then use the average of the Hessians at these indices as an (unbiased) estimator of $\nabla^2 f(x_k)$

$$B_k := \frac{1}{|\mathcal{S}_{B,k}|} \sum_{i \in \mathcal{S}_{B,k}} \nabla^2 f_i(w_k). \tag{4.28}$$

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

Lemma 4.9 (Hessian deviation bound) *Let the sub-sampled Hessian B be defined as in Eq. (4.28). As long as $\varepsilon \leq 4\kappa_g$, we have with probability $(1-\delta)$ that*

$$\|B(w_k) - H(w_k)\| \leq 4\kappa_g \sqrt{\frac{\log((2d)/\delta)}{|\mathcal{S}_{B,k}|}}. \tag{4.29}$$

Proof: Bernstein's Inequality holds as $f \in C^2$ and thus the Hessian is symmetric by Schwarz's Theorem. Since the expectation of the random matrix needs to be zero, we center the individual Hessians

$$X_i = H_i(w_k) - H(w_k),\ i = 1, ..., |\mathcal{S}_{B,k}|$$

and note that now from the Lipschitz continuity of g (A4.1) we can deduce

$$\|X_i\|_2 \leq 2\kappa_g, i = 1...|\mathcal{S}_{B,k}| \text{ and } \|X_i^2\|_2 \leq 4\kappa_g^2, i = 1...|\mathcal{S}_{B,k}|.$$

Hence, for $\varepsilon \leq 4\kappa_g$ we are in the *small deviation* regime of Bernstein's bound with a sub-gaussian tail. Then, we may plug

$$\frac{1}{|\mathcal{S}_{B,k}|} \sum_{i=1}^{|\mathcal{S}_{B,k}|} X_i = B(w_k) - H(w_k)$$

into (4.12), to get

$$P(\|B(w_k) - H(w_k)\| \geq \varepsilon) \leq 2d \cdot \exp\left(-\frac{\varepsilon^2 |\mathcal{S}_{B,k}|}{16\kappa_g^2}\right). \tag{4.30}$$

Finally, we shall require the probability of a deviation of ε or higher to be lower than some $\delta \in (0, 1]$

$$\begin{aligned} & 2d \cdot \exp\left(-\frac{\varepsilon^2 |\mathcal{S}_{B,k}|}{16\kappa_g^2}\right) \overset{!}{\leq} \delta \\ \Leftrightarrow\ & -\frac{\varepsilon^2 |\mathcal{S}_{B,k}|}{16\kappa_g^2} \overset{!}{\leq} \log(\delta/(2d)) \\ \Leftrightarrow\ & \varepsilon \overset{!}{\geq} 4\kappa_g \sqrt{\frac{\log((2d)/\delta)}{|\mathcal{S}_{B,k}|}}, \end{aligned} \tag{4.31}$$

which is equivalent to $\|B(w_k) - H(w_k)\|$ staying within this particular choice of ε with probability $(1-\delta)$, generally perceived as *high probability*.

□

Again, this Lemma can directly be used to derive a Hessian sampling condition that is guaranteed to satisfy the sufficient agreement condition (A4.2) with high probability.

Theorem 4.10 (Hessian Sampling) *If*

$$|\mathcal{S}_{B,k}| \geq \frac{16\kappa_g^2 \log((2d)/\delta)}{(C\|s_k\|)^2},\ C \geq 0,\ \textit{and}\ \forall k \geq 0 \tag{4.32}$$

then B_k satisfies the strong agreement condition A4.2 with probability $(1-\delta)$.

Proof: Since $\|A\mathbf{v}\| \leq \|A\|_{op}\|\mathbf{v}\|$ for every $\mathbf{v} \in V$ we have for the choice of the spectral matrix norm and euclidean vector norm that any B_k that satisfies $\|(B(w_k) - H(w_k))\| \leq C\|s_k\|$ also satisfies the sufficient agreement condition Asm. 4.2. Furthermore,

$$\begin{aligned} &\|(B - H(w_k))\| \leq C\|s_k\| \\ \overset{w.h.p.}{\Leftrightarrow}\ &4\kappa_g\sqrt{\frac{\log((2d)/\delta)}{|\mathcal{S}_{B,k}|}} \leq C\|s_k\| \\ \Leftrightarrow\ &|\mathcal{S}_{B,k}| \geq \frac{16\kappa_g^2 \log((2d)/\delta)}{(C\|s_k\|)^2}, \quad C > 0. \end{aligned} \tag{4.33}$$

□

As expected, the required sample sizes grow in the problem dimensionality d and in the Lipschitz constants κ_f and κ_g. Note that it might be possible to derive a less restrictive sampling condition that satisfy A4.2 since condition (4.33) is based on the worst case bound $\|A\mathbf{v}\| \leq \|A\|_{op}\|\mathbf{v}\|$ which indeed only holds with equality if $\mathbf{v}$ happens to be (exactly in the direction of) the largest eigenvector of A.

Finally, we shall restate a Lemma from [Cartis et al., 2011] which illustrates that the stepsize goes to zero and hence the sample size to n as the algorithm converges.

Lemma 4.11 *Let $\{f(w_k)\}$ be bounded below by some $f_{\text{inf}} > -\infty$. Also, let s_k satisfy A3.7 and σ_k be bounded below by some $\sigma_{\text{inf}} > 0$. Then we have for all successful iterations that*

$$\|s_k\| \to 0, \text{ as } k \to \infty. \tag{4.34}$$

The proof can be found in [Cartis et al., 2011] Section 5. Consequently, the sample sizes used in SCR must approach n as the algorithm converges and thus we have

$$g \to \nabla f \text{ as well as } B \to H \text{ as } k \to \infty. \tag{4.35}$$

On the left hand side of Figure 4.1 we illustrate the Hessian sample sizes that result when applying SCR with a practical version of Theorem 4.10 to the *higgs* dataset[3]. On the right, we benchmark our algorithm to the deterministic as well as two stochastic version of ARC with *linearly* and *exponentially* increasing sample sizes.

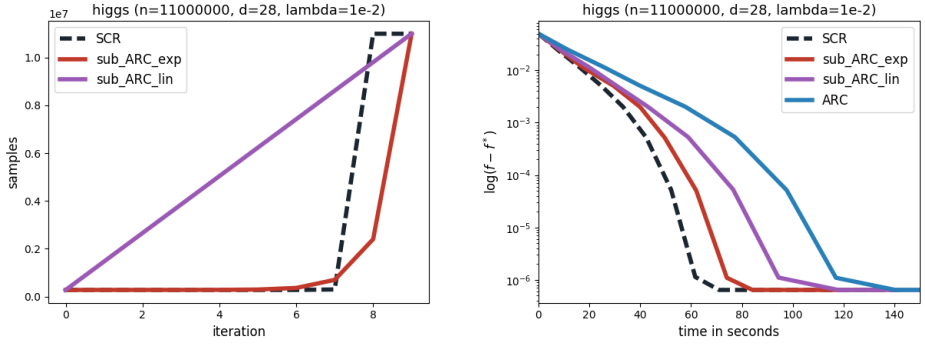


Figure 4.1: Suboptimality and sample size for different cubic regularization methods on *higgs*

The difference in runtime is not particularly stunning but note that both the linear- and the exponential sampling scheme were carefully fine tuned to reach the full sample size at the very last iteration. In general however, the total number of iterations an algorithm may take is obviously unknown and it is thus a clear advantage of SCR that no such sample size tuning is needed ex-ante.

4.3 Convergence Analysis

We shall now establish the crucial properties that ensure global, as well as fast local convergence and improve the worst-case complexity of fully

[3] see Chapter 5 for details

stochastic cubic regularization 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 this framework, which give rise to a good relation between regularization and stepsize. In this section we do not assume any specific sampling scheme. Instead, all we require is that the stochastic estimates are unbiased and satisfy the sufficient agreement conditions (4.2) and (4.3). Yet, in the case that Theorem (4.8) and (4.10) are satisfied by the applied sampling scheme the resulting method satisfies these requirements with high probability. Hence, the following convergence results hold with high probability for the SCR version that we implement in Chapter 5.

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 4.12 (Boundedness of σ_k) *Let A4.1, A4.2 and A4.3 hold. Then*

$$\sigma_k \in [\sigma_{\text{inf}}, \sigma_{\text{sup}}], \ \forall k \geq 0, \tag{4.36}$$

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

$$\sigma_{sup} := \left\{ \sigma_0, \frac{3}{2}\gamma_2(2M + C + \kappa_g) \right\}. \tag{4.37}$$

Proof: The lower bound σ_{inf} follows directly from Step 7 in the algorithm design (see Algorithm 1). Regarding the upper bound, the constant σ_0 accounts for the start value of the penalty parameter. Let us now show that as soon as some $\sigma_k > 3(\frac{2M+C+\kappa_g}{2})$, the iteration is very successful and $\sigma_{k+1} < \sigma_k$. Finally, γ_2 allows for σ_k being "close to" the successful threshold, but increased one last time.

Any iteration with $f(w_k + s_k) \leq m(s_k)$ yields a $\rho_k \geq 1 \geq \eta_2$ and is thus very successful. From a second-order Taylor approximation of $f(w_k + s_k)$ around

w_k we have:

$$\begin{aligned} f(w_k + s_k) - m_k(s_k) &= (\nabla f(w_k) - g(w_k))^\intercal s_k + \frac{1}{2} s_k^\intercal \left(H(w_k + \tau s_k) - B_k\right) s_k \\ &\quad - \frac{\sigma}{3}\|s_k\|^3 \\ &\leq e_k^\intercal s_k + \frac{1}{2}\|s_k\|^2 \|H(w_k + \tau s_k) - H(x)\| \\ &\quad + \frac{1}{2}\|H(w_k) - B_k\|\|s_k\| - \frac{\sigma_k}{3}\|s_k\|^3 \\ &\leq \|e_k\|\|s_k\| + \left(\frac{C + \kappa_g}{2} - \frac{\sigma_k}{3}\right)\|s_k\|^3 \\ &\leq M\|s_k\|^3 + \left(\frac{C + \kappa_g}{2} - \frac{\sigma_k}{3}\right)\|s_k\|^3 \\ &= \left(\frac{2M + C + \kappa_g}{2} - \frac{\sigma_k}{3}\right)\|s_k\|^3, \end{aligned} \tag{4.38}$$

where we applied the sufficient agreement conditions (4.1) & (4.3), Cauchy-Schwarz's inequality as well as the Lipschitz continuity of H. Requiring the right hand side to be non-positive and solving for σ_k gives the desired result.

□

Furthermore, for any successful iteration the objective decrease can be directly linked to the model decrease via the step acceptance criterion in Eq. (3.6). The latter, in turn, can be shown to be lower bounded by the stepsize which combined gives the following result.

Lemma 4.13 (Sufficient function decrease) *Suppose that s_k satisfies A3.7. Then, for all successful iterations $k \geq 0$*

$$\begin{aligned} f(w_k) - f(w_{k+1}) &\geq \eta_1 (f(w_k) - m(s_k)) \\ &\geq \frac{1}{6}\eta_1 \sigma_{\text{inf}} \|s_k\|^3. \end{aligned} \tag{4.39}$$

Proof: By definition of the stochastic model $m_k(s_k)$ we have

$$\begin{aligned} f(w_k) - m_k(s_k) &= -s_k^\intercal g(w_k) - \frac{1}{2} s_k^\intercal B_k s_k - \frac{1}{3}\sigma_k \|s_k\|^3 \\ &= \frac{1}{2} s_k^\intercal B_k s_k + \frac{2}{3}\sigma_k \|s_k\|^3 \\ &\geq \frac{1}{6}\sigma_k \|s_k\|^3, \end{aligned} \tag{4.40}$$

where we applied equation (3.53) first and equation (3.54) secondly.

□

Finally, the termination criterion TC also guarantees step sizes that do not become too small compared to the respective gradient norm. However, before we can prove the lower bound on the stepsize $\|s_k\|$ we must first transfer the rather technical result from Lemma 4.6 in [Cartis et al., 2011] to our framework of stochastic gradients.

Lemma 4.14 *Let $f \in C^2$, Lipschitz continuous gradients (A4.1) and TC (A3.9) hold. Then, for each (very-) successful k, we have*

$$(1-\kappa_\theta)\|\nabla f(w_{k+1})\| \leq \sigma_k \|s_k\|^2 + \|s_k\|\zeta_k, \tag{4.41}$$

with $\zeta_k = \zeta_{k,1} + \zeta_{k,2}$ and

$$\begin{aligned} \zeta_{k,1} &:= \|\int_0^1 (H(w_k + ts_k) - H(w_k))dt\| + \frac{\|(H(w_k) - B_k)s_k\|}{\|s_k\|} \\ \zeta_{k,2} &:= \kappa_\theta \kappa_g \|s_k\| + (1 + \kappa_\theta \kappa_g)\frac{\|e_k\|}{\|s_k\|} \end{aligned} \tag{4.42}$$

with $\kappa_\theta \in (0,1)$ as in TC (3.68).

Proof: We shall by noting that $w_{k+1} = w_k + s_k$ which is why we can write

$$\begin{aligned} \|\nabla f(w_k + s_k)\| &\leq \|\nabla f(w_k + s_k) - \nabla m_k(s_k)\| + \|\nabla m_k(s_k)\| \\ &\leq \underbrace{\|\nabla f(w_k + s_k) - \nabla m_k(s_k)\|}_{(a)} + \underbrace{\theta_k \|g_k(w_k)\|}_{(b)}, \end{aligned} \tag{4.43}$$

where the last inequality results from TC. Now, let us find bounds on each of the above summands:

(a) By (3.2) we have

$$\|\nabla f(w_k + s_k) - \nabla m_k\| = \|\nabla f(w_k + s_k) - g_k(w_k) - B_k s_k - \sigma_k s_k \|s_k\|\|. \tag{4.44}$$

We can rewrite the right-hand side by a Taylor expansion of $\nabla f(w_k + s_k)$ around w_k to get

$$(4.44) = \|\nabla f(w_k) + \int_0^1 H(w_k + \tau s_k)s_k d\tau - g_k(w_k) - B_k s_k - \sigma_k s_k \|s_k\|\|, \tag{4.45}$$

for some $\tau \in (0,1)$. Contrary to the case of deterministic gradients, the first and third summand no longer cancel out. Applying the triangle inequality

repeatedly, we thus get the error term e_k in the final bound on (a):

$$\begin{aligned}\|\nabla f(w_k + s_k) - \nabla m_k\| &\leq \|\int_0^1 H((w_k + ts_k) - B_k)s_k dt\| + \sigma_k\|s_k\|^2 \\ &\quad + \|\nabla f(w_k) - g_k(w_k)\| \\ &\leq \|\int_0^1 H((w_k + ts_k)dt - H(w_k)\| \cdot \|s_k\| \\ &\quad + \|(H(w_k) - B_k)s_k\| + \sigma_k\|s_k\|^2 + \|e_k\|.\end{aligned} \tag{4.46}$$

(b) To bound the second summand, we can write

$$\begin{aligned}\|g(w_k)\| &\leq \|\nabla f(w_k)\| + \|e_k\| \\ &\leq \|\nabla f(w_k + s_k)\| + \|\nabla f(w_k) - \nabla f(w_k + s_k)\| + \|e_k\| \\ &\leq \|\nabla f(w_k + s_k)\| + \kappa_g\|s_k\| + \|e_k\|.\end{aligned} \tag{4.47}$$

Finally, using the definition of θ_k as in (3.68) (which also gives $\theta_k \leq \kappa_\theta$) and summing up Eq. (4.46) and (4.47) gives (4.41) which proves the assertion.

□

Lemma 4.15 (Lower bound on stepsize) *Let Asm. 4.1, Asm. 4.2 and Asm. 4.3 hold. Furthermore, assume TC and suppose that $w \to w^*$, as $k \to \infty$. Then, for all sufficiently large successful iterations s_k satisfies*

$$\|s_k\| \geq \kappa_s \sqrt{\|\nabla f(w_{k+1})\|} \tag{4.48}$$

where κ_s is a positive constant

$$\kappa_s \leq \sqrt{\frac{1 - \kappa_\theta}{\frac{1}{2}\kappa_H + (1 + \kappa_\theta\kappa_g)M + C + \sigma_{\sup} + \kappa_\theta\kappa_g}}. \tag{4.49}$$

Proof: The conditions of Lemma 4.14 are satisfied. By multiplying $\zeta_k\|s_k\|$ out in equation (4.41) we get

$$\begin{aligned}(1 - \kappa_\theta)\|\nabla f(w_{k+1})\| \leq &\|\int_0^1 (H(w_k + \tau s_k) - H(w_k))d\tau\|\|s_k\| \\ &+ \|(H(w_k) - B_k)s_k\| + \kappa_\theta\kappa_g\|s_k\|^2 \\ &+ (1 + \kappa_\theta\kappa_g)\|e_k\| + \sigma_k\|s_k\|^2.\end{aligned} \tag{4.50}$$

Now, applying the sufficient agreement conditions (4.3) and (4.1) as well as the Lipschitz continuity of H we can rewrite this as

$$(1 - \kappa_\theta)\|\nabla f(w_{k+1})\| \leq (\frac{1}{2}\kappa_H + C + (1 + \kappa_\theta\kappa_g)M + \sigma_{\sup} + \kappa_\theta\kappa_g)\|s_k\|^2, \tag{4.51}$$

for all sufficiently large, successful k. Solving for the step size $\|s_k\|$ provides the above results.

□

4.3.2 Local Convergence

Since the models m_k depend on random quantities, the iterates generated by these random models constitute a sequence of random variables themselves. However, given that the stochastic gradients and Hessians are sampled uniformly and independently, the approximation error vanishes in expectation, i.e. $\mathbb{E}[g_k] = \nabla f(w_k)$ and $\mathbb{E}[B_k] = \nabla^2 f(w_k)$. Hence, the local convergence result we present in this section refers to the norm of the expectation of $\{w_k\}$, commonly named as convergence in expectation (See Definition A.3).

Yet, before we can study the convergence rate of SCR in a locally convex neighbourhood of a local minimizer w_* we first need to establish three crucial properties:

a) a lower bound on $\|s\|$ that depends on $\|g_k\|$.

b) an upper bound on $\|s\|$ that depends on $\|g_{k+1}\|$.

c) conditions under which all steps are eventually very successful.

With this at hand we will be able to relate $\|g_{k+1}\|/\|g_k\|$, show that (in expectation) this ratio eventually goes to zero at a quadratic rate and conclude from a Taylor expansion around g that the iterates themselves converge as well (see Lemma A.6).

We have already established (a) in Lemma 4.15 so let us turn our attention directly to b):

Lemma 4.16 (Upper bound on stepsize) *Suppose that s_k satisfies (3.53) and that the Rayleigh coefficient $R_k(s_k)$ (Def. A.9) is positive, then*

$$\|s_k\| \leq \frac{1}{R_k(s_k)}\|g_k\| = \frac{1}{R_k(s_k)}\|\nabla f(w_k) + e_k\| \leq \frac{1}{R_k(s_k)}(\|\nabla f(w_k)\| + \|e_k\|) \tag{4.52}$$

Proof: Given the above assumptions we can rewrite (3.53) as follows

$$R_k(s_k)\|s_k\|^2 = -s_k^T g_k - \sigma_k\|s_k\|^3 \leq \|s_k\|\|g_k\|, \tag{4.53}$$

where we used Cauchy-Schwarz inequality as well as the fact that $\sigma_k > 0$, $\forall k$. Solving 4.53 for $\|s_k\|$ gives (4.52).

□

Now that we have both stepsize bounds note that they only hold for *sufficiently large successful iterations*. Thus, we shall now establish c), namely that, when converging, all SCR iterations are indeed very successful asymptotically.

Lemma 4.17 (Eventually successful iterations) *Let $f \in C^2$, ∇f uniformly continuous and B_k bounded above. Let B and g satisfy the agreement conditions Asm. 4.2 and Asm. 4.3, as well as s_k satisfy (3.53). Furthermore, let*

$$w_k \to w_*, \text{ as } k \to \infty, \tag{4.54}$$

with $\nabla f(w_*) = 0$ and $H(w_*)$ positive definite. Then there exists $R_{min} > 0$ such that for all k sufficiently large

$$R_k(s_k) \geq R_{\min}. \tag{4.55}$$

Furthermore, in expectation all iterations are eventually very successful.

Proof: Since f is continuous, the limit (4.54) implies that $\{f(w_k)\}$ is bounded below. Since $H(w_*)$ is positive definite per assumption, so is $H(w_k)$ for all k sufficiently large. Therefore, there exists a constant $R_{\min}$ such that

$$\frac{s_k^\intercal H(w_k) s_k}{\|s_k\|^2} > R_{\min} > 0, \text{ for all } k \text{ sufficiently large.} \tag{4.56}$$

In order to show that asymptotically all iterations k are very successful, we need to ensure that the following quantity r_k eventually becomes negative:

$$r_k := \underbrace{f(w_k + s_k) - m(s_k)}_{(i)} + (1 - \eta_2) \underbrace{(m(s_k) - f(w_k))}_{(ii)}, \tag{4.57}$$

where $\eta_2 \in (0, 1)$ is the "very successful" threshold.

(i) By a (second-order) Taylor approximation around $f(w_k)$ and applying the Cauchy-Schwarz inequality, we have:

$$\begin{aligned} f(w_k + s_k) - m(s_k) =& (\nabla f(w) - g_k)^\intercal s_k + \frac{1}{2} s_k^\intercal ((H(w_k + \tau s_k) - B_k) s_k - \frac{\sigma_k}{3} \|s\|^3 \\ \leq& \|e_k\| \|s_k\| + \frac{1}{2} \|((H(w_k + \tau s_k) - B_k) s_k\| \|s_k\|, \end{aligned} \tag{4.58}$$

where the term $\|e_k\|\|s_k\|$ is extra compared to the case of deterministic gradients.

(ii) Regarding the second part we note that if s_k satisfies (3.53), we have by the definition of R_k and equation (4.55) that

$$\begin{aligned} f(w_k) - m_k(s_k) =& \frac{1}{2} s_k^\intercal B s_k + \frac{2}{3} \sigma_k \|s_k\|^3 \\ \geq& \frac{1}{2} R_{\min} \|s_k\|^2, \end{aligned} \tag{4.59}$$

which negated gives the desired bound on (ii). All together, the upper bound on r_k is written as

$$r_k \leq \frac{1}{2}\|s_k\|^2 \left(\frac{2\|e_k\|}{\|s_k\|} + \frac{\|((H(w_k + \tau s_k) - B_k)s_k\|}{\|s_k\|} - (1 - \eta_2)R_{\min} \right). \quad (4.60)$$

Let us add and subtract $H(w_k)$ to the second summand and take the expectation of r_k to find

$$\mathbb{E}[r_k] \leq \frac{1}{2}\|\mathbb{E}[s_k]\|^2 \left(\frac{\|((H(w_k + \tau\mathbb{E}[s_k]) - H(w_k))\mathbb{E}[s_k]\|}{\|\mathbb{E}[s_k]\|} - (1 - \eta_2)R_{\min} \right), \quad (4.61)$$

where we furthermore applied $\mathbb{E}[e_k] = 0$ & $\mathbb{E}[H_k - B_k] = 0$ as well as the Cauchy-Schwarz inequality.

Since $\tau \in [0, 1]$ we have that $\|w_k + \tau\mathbb{E}[s_k]\| \leq \|w_k + \mathbb{E}[s_k]\|$ but from (4.16) we know that $\mathbb{E}[s_k] \to 0$ whenever $\nabla f(w_k) \to 0$. Hence, $H(w_k + \tau s_k)$ and $H(w_k)$ eventually agree. Finally, $\eta_2 < 1$ and $R_{\min} > 0$ such that $\mathbb{E}[r_k]$ is negative for all k sufficiently large, which implies that every such iteration is very successful.

□

Theorem 4.18 (Quadratic local convergence in expectation) *Let Asm. 4.1, Asm. 4.2 and Asm. 4.3 hold. Furthermore, let s_k satisfy Asm. 3.7 and*

$$w_k \to w^* \text{ as } k \to \infty, \quad (4.62)$$

where $H(w^*)$ is positive definite. Moreover, assume the stopping criterion TC (Asm 3.9). Then,

$$\frac{\|\mathbb{E}[w_{k+1} - w^* | w_k]\|}{\|w_k - w^*\|^2} \leq c,\ c > 0 \text{ as } k \to \infty. \quad (4.63)$$

That is, w_k converges in expectation quadratically to w^* as $k \to \infty$.

Proof: We will first derive the convergence result carrying potential approximation errors arising from the stochastic gradients and Hessian over along the way.

From Lemma 4.12 we have $\sigma_k \leq \sigma_{sup}$. Furthermore, all assumptions needed for the step size bounds of Lemma 4.15 and 4.16 hold. Finally, Lemma 4.17 gives that all iterations are eventually successful (in expectation). Thus, we

can combine the upper (4.52) and lower (4.48) bound on the stepsize for all k sufficiently large to obtain

$$\frac{1}{R_{\min}}(\|\nabla f(w_k)\| + \|e_k\|) \geq \|s_k\| \geq \kappa_s \sqrt{\|\nabla f(w_{k+1})\|} \tag{4.64}$$

which we can solve for the gradient norm ratio

$$\frac{\|\nabla f(w_{k+1})\|}{\|\nabla f(w_k)\|^2} \leq \left(\frac{1}{R_{\min}\kappa_s}\left(1 + \frac{\|e_k\|}{\|\nabla f(w_k)\|}\right)\right)^2. \tag{4.65}$$

Consequently, as long as the right hand side of (4.65) stays below infinity, i.e. $\|e_k\|/\|\nabla f(w_k)\| \not\to \infty$, we have quadratic convergence of the gradient norms. Let us now take the expectation of (4.65)

$$\begin{aligned}\frac{\|\mathbb{E}\left[\nabla f(w_{k+1})\right]\|}{\|\mathbb{E}\left[\nabla f(w_k)\right]\|^2} &\leq \left(\frac{1}{R_{\min}\mathbb{E}\left[\kappa_s\right]}\left(1 + \frac{\|\mathbb{E}\left[e_k\right]\|}{\|\nabla f(w_k)\|}\right)\right)^2 \\ &= \left(\frac{1}{R_{\min}\mathbb{E}\left[\kappa_s\right]}\right)^2,\end{aligned} \tag{4.66}$$

where (from Eq. (4.49):

$$\mathbb{E}\left[\kappa_s\right] \leq \sqrt{\frac{1-\kappa_\theta}{\frac{1}{2}\kappa_H + \sigma_{\sup} + \kappa_\theta\kappa_g}}. \tag{4.67}$$

Thus $\mathbb{E}\left[\kappa_s\right]$ is bounded above by a constant and since furthermore R_{min} is a positive constant itself we have proven that the right hand side of equation (4.65) is bounded in expectation. The convergence of the iterates (4.69) follows from Lemma A.6, which proves the assertion.

□

Note that the need of using the expected values in Lemma 4.17 and Theorem 4.18 solely arises due to the inexactness of g which makes it impossible to upper bound s_k in terms of ∇f. However, the quadratic convergence result can be obtained in a deterministic fashion when only the Hessians are subsampled.

Theorem 4.19 (Quadratic local convergence) *Let Asm. 4.1 and Asm. 4.2 hold and assume $g_k = \nabla f(w_k)$. Let s_k satisfy Asm. 3.7 and*

$$w_k \to w^*, \text{ as } k \to \infty, \tag{4.68}$$

where $H(w^*)$ is positive definite. Moreover, assume the stopping criterion TC (Asm. 3.9). Then,

$$\frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|^2} \leq c,\ c > 0 \text{ as } k \to \infty. \tag{4.69}$$

That is, w_k converges quadratically to w^* as $k \to \infty$.

Proof: Follow the proofs of Lemma 4.17 and Theorem 4.19 to Eq. (4.61) and Eq. (4.65) respectively and use the fact that in the case of deterministic gradients we have $e_k = g_k - \nabla f(w_k) = 0$.

□

4.3.3 Global Convergence

First order critical points

The preliminary results Lemma 4.13 and 4.15 allow us to lower bound the function decrease of a successful step in terms of the *full* gradient ∇f_{k+1}. Combined with Lemma 4.12, this enables us to give a *deterministic* global convergence guarantee while using only *stochastic* first order information[4].

Theorem 4.20 (Convergence to 1st-order Critical Points) *Let Asm. 3.7, Asm. 4.1, Asm. 4.2 and Asm. 4.3 hold. Furthermore, let $\{f(w_k)\}$ be bounded below by some $f_{\text{inf}} > -\infty$. Then*

$$\lim_{k \to \infty} \|\nabla f(w_k)\| = 0 \tag{4.70}$$

Proof: We will consider two cases regarding the number of successful steps for this proof.

Case (i): SCR takes only finitely many successful steps. Hence, we have some index k_0 which yields the very last successful iteration and all further iterates stay at the same point w_{k_0+1}. That is $w_{k_0+1} = w_{k_0+i},\ \forall\, i \geq 1$. Let us assume that $\|\nabla f(w_{k_0+1})\| = \varepsilon > 0$, then

$$\|\nabla f(w_k)\| = \varepsilon,\ \forall\, k \geq k_0 + 1. \tag{4.71}$$

[4]Note that this result can also be proven without Lipschitz continuity of H and less strong agreement conditions as done in Corollary 2.6 in [Cartis et al., 2011].

Now, since all iterations $k \geq k_0 + 1$ are unsuccessful, σ_k increases by γ in each of these iterations such that

$$\sigma_k \to \infty \text{ as } k \to \infty. \tag{4.72}$$

However, this is in contradiction with Lemma 4.12, which states that σ_k is bounded above. Hence, the above assumption cannot hold and we have $\|\nabla f(w_{k_0+1})\| = \|\nabla f(w^*)\| = 0$.

Case (ii): SCR takes infinitely many successful steps. While unsuccessful steps keep $f(w_k)$ constant, (very) successful steps strictly decrease $f(w_k)$ and thus the sequence $\{f(w_k)\}$ is monotonically decreasing. In addition, it is bounded below per assumption and thus the objective values converge

$$f(w_k) \to f_{\text{inf}}, \text{ as } k \to \infty. \tag{4.73}$$

All requirements of Lemma 4.13 and Lemma 4.15 hold and we thus can combine Eq. (4.39) and Eq. (4.48) to write

$$f(w_k) - f_{\text{inf}} \geq f(w_k) - f(w_{k+1}) \geq \frac{1}{6}\eta_1 \sigma_{\text{inf}} \kappa_s^3 \|\nabla f(w_{k+1})\|^{3/2}. \tag{4.74}$$

Since $(f(w_k) - f_{\text{inf}}) \to 0$ as $k \to \infty$ and $\sigma_{\text{inf}} > 0, \eta_1 > 0$ as well as $\kappa_s^3 > 0$ (because $\sigma_{\text{sup}} < \infty$), we must have $\|\nabla f(w_k)\| \to 0$, which proves the assertion.

□

second-order critical points

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 much of the proof as it can be found in Section 5 of Cartis et al. [2011]. We shall nevertheless state the adaptions to our framework here for the sake of completeness.

Theorem 4.21 (Second-order global convergence) *Let Asm. 4.1, Asm. 4.2 and Asm. 4.3 hold. Furthermore, let $\{f(w_k)\}$ bounded below by f_{inf} and let 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 Q_k. If $B \to H$ asymptotically, any subsequence of negative leftmost eigenvalues $\{\lambda_{\min}(Q_k^T H(w_k) Q_k)\}$ converges to zero for sufficiently large, successful iterations. Hence*

$$\lim_{k \, succ. \, k \to \infty} \inf \lambda_{\min}(Q_k^T H(w_k) Q_k) \geq 0. \tag{4.75}$$

Finally, if 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 $\{w_k\}$ is second-order critical (provided such a limit point exists).

Sketch of proof: Regarding the subspace minimization we can use the results elaborated thoroughly in Lemma 3.8. Thus, for all $s \in \mathcal{L}$ we have $s = Q_k u$, $u \in \mathbb{R}^l$ and any global minimizer s_k of m_k in $\mathcal{L}_k$ satisfies Eq. (3.64), which is

$$Q_k^\intercal B_k Q_k + \sigma_k \|Q_k u_k\| I = Q_k^\intercal B_k Q_k + \sigma_k \|s_k\| I \succeq 0. \tag{4.76}$$

By applying Lemma 4.12 we can reformulate this for all iterations $k \geq 0$ with a negative definite $Q_k^\intercal B_k Q_k$ to

$$\sigma_{\sup} \|s_k\| \geq \sigma_k \|s_k\| \geq -\lambda_{\min}(Q_k^\intercal B_k Q_k) = |\lambda_{\min}(Q_k^\intercal B_k Q_k)| \tag{4.77}$$

which for accepted steps, whole length eventually converges to zero according to Lemma 4.11), implies

$$\lim_{k\, succ.\, k\to\infty} \inf \lambda_{\min}(Q_k^\intercal B(w_k) Q_k) \geq 0. \tag{4.78}$$

Furthermore, if the Hessian and its approximation finally agree, i.e. $\|H_k - B_k\| \to 0$, whenever $\|g_k\| \to 0$ we have

$$\lim_{k\, succ.\, k\to\infty} \inf \lambda_{\min}(Q_k^\intercal H_k Q_k) \geq 0, \tag{4.79}$$

which proves the first part of the assertion, namely that any subsequence of negative leftmost eigenvalues $\{\lambda_{\min}(Q_k^\intercal H(w_k) Q_k)\}$ converges to zero as for sufficiently large successful iterations (Eq. (4.75)).

Note that $Q_k^\intercal Q_k = I$, since the columns form an *orthonormal* basis of $\mathcal{L}_k$. However, as long as $l < n$, $Q_k Q_k^\intercal \neq I$ since the n row-vectors of Q_k cannot be linearly independent[5] in $\mathbb{R}^l$. Thus, in order to ensure that any limit point of $\{x_k\}$ is actually second-order critical in $\mathbb{R}^d$, we need Q_k to become a full orthogonal basis of $\mathbb{R}^d$, i.e. $l = d$. Because ony then, the eigenvalues of $Q_k^\intercal H(w_k) Q_k$ and $H(w_k)$ coincide as can be seen in Lemma A.8.

□

Note that, when the Krylov subspace minimization routine from Section 3.2.4 is applied, Q_k can indeed be expected to become a full orthogonal basis

[5] In fact, $Q_k Q_k^\intercal$ constitutes the orthogonal projection P onto $col(Q_k)$ since $P = Q_k (Q_k^\intercal Q_k)^{-1} Q_k^\intercal = Q_k Q_k^\intercal$

provided the gradient is not orthogonal to any eigenvector of B_k [Cartis et al., 2011]. Furthermore, SCR versions that sample according to Theorem 4.10 the Hessian and its approximations are guaranteed to agree eventually because of Lemma 4.11 which implies Eq. (4.35), such that the above derived results holds with high probability.

4.3.4 Worst-case Complexity

For the worst-case analysis we shall establish the two disjunct 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. Furthermore, let us impose the following weak restriction on the penalty parameter decrease for successful iterations

$$\sigma_{k+1} \geq \gamma_3 \sigma_k, \text{ for some } \gamma_3 \in (0, 1] \text{ and all } k \in \mathcal{S}_\infty \tag{4.80}$$

As stated in Lemma 4.12 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 4.22 (Number of unsuccessful iterations) *For any fixed $j \geq 0$, next to Eq. (4.80) let all assumptions of Lemma 4.12 hold. Then we have that*

$$|\mathcal{U}_j| \leq \left\lceil (|\mathcal{S}_j| + 1) \frac{\log(\sigma_{\text{sup}}) - \log(\sigma_{\text{inf}})}{\log(\eta_1)} \right\rceil. \tag{4.81}$$

Proof: From the algorithm construction of SCR we have

$$\gamma_1 \sigma_k \leq \sigma_{k+1}, \text{ for all } k \in \mathcal{U}_j \tag{4.82}$$

and since equation (4.80) holds for all $k \in \mathcal{S}_j$ per assumption, we can deduce inductively that

$$\sigma_0 \gamma_3^{|\mathcal{S}_j|} \gamma_1^{|\mathcal{U}_j|} \leq \sigma_j. \tag{4.83}$$

Because $\sigma_{\text{inf}} \leq \sigma_{\text{sup}}$ we may choose $\eta_3 := \sigma_{\text{inf}}/\sigma_{\text{sup}} \in (0, 1]$. This, and Lemma 4.12 yield

$$\log(\sigma_{\text{inf}}) + |\mathcal{S}_j| \log(\frac{\sigma_{\text{inf}}}{\sigma_{\text{sup}}}) + |\mathcal{U}_j| \log(\mu_1) \leq \log(\sigma_{\text{sup}}), \tag{4.84}$$

which can be rearranged to give the desired upper bound on $\mathcal{U}_j$.

□

Regarding the number of successful iterations we have already established the two key ingredients: (i) a sufficient function decrease in each successful iteration (Lemma 4.13) and (ii) a step size that does not become too small compared to the respective gradient norm (Lemma 4.15), which is essential for driving the latter below ε at a fast rate. Combined they give rise to the guaranteed function decrease for successful iterations

$$f(w_k) - f(w_{k+1}) \geq \frac{1}{6}\eta_1 \sigma_{\text{inf}} \kappa_s^3 \left\| \nabla f(w_{k+1}) \right\|^{3/2}, \tag{4.85}$$

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

Theorem 4.23 (First-order worst-case complexity) *Let Asm. 3.7, Asm. 4.1, Asm. 4.2 and Asm. 4.3 hold. Furthermore, let $\{f(w_k)\}$ bounded below by f_{inf} and the termination critertion TC (Asm. 3.9) be applied . Then, for $1 \geq \varepsilon > 0$ the total number of iterations that stochastic cubic regularization methods take to generate the first iterate j with $\|\nabla f(w_{j+1})\| \leq \varepsilon$ is bounded as follows:*

$$j \leq \left\lceil (1+\kappa_i)(2+\kappa_j)\varepsilon^{-3/2} \right\rceil, \tag{4.86}$$

where

$$\kappa_i = \frac{6(f(w_0) - f_{\text{inf}})}{\eta_1 \sigma_{\text{inf}} \kappa_s^3} \text{ and } \kappa_j = \frac{\log(\sigma_{\text{sup}}) - \log(\sigma_{\text{inf}})}{\log(\eta_1)} \tag{4.87}$$

Proof: Since f is bounded below for all iterates and $\|f(w_{k+1})\| > \varepsilon$ for all $k < j$, we get by summing over all iterations in $\mathcal{S}_{j-1}$ and applying inequality (4.85) that

$$\begin{aligned} f(w_0) - f_{\text{inf}} &\geq \sum_{k \in \mathcal{S}_{j-1}} f(w_k) - f(w_{k+1}) \\ &\geq \frac{1}{6}\eta_1 \sigma_{\text{inf}} \kappa_s^3 \sum_{k \in \mathcal{S}_{j-1}} \varepsilon^{3/2} \\ &= \frac{1}{6}\eta_1 \sigma_{\text{inf}} \kappa_s^3 \left|\mathcal{S}_{j-1}\right| \varepsilon^{3/2}. \end{aligned} \tag{4.88}$$

Solving this inequality for $|\mathcal{S}_{j-1}|$ and noting that $|\mathcal{S}_{j-1}| \in \mathbb{N}$ gives

$$\left|\mathcal{S}_{j-1}\right| \leq \left\lfloor \kappa_i \varepsilon^{-3/2} \right\rfloor. \tag{4.89}$$

Of course, that the last iteration j itself must be successful such that we can state the following bound on the overall number of successful iterations

$$\left|\mathcal{S}_j\right| \leq \left\lfloor \kappa_i \varepsilon^{-3/2} \right\rfloor + 1 = \left\lceil \kappa_i \varepsilon^{-3/2} \right\rceil, \tag{4.90}$$

where κ_i is defined as in (4.87).Furthermore, σ_k is bounded since all conditions of Lemma 4.12 hold and we can thus apply the upper bound on the number of unsuccessful iterations from Lemma 4.22 to find that

$$\begin{aligned} |S_j| + |U_j| \leq & |S_j| + \lceil (|S_j| + 1)\kappa_j \rceil \\ & \leq \left\lceil \kappa_i \varepsilon^{-3/2} \right\rceil + \left\lceil (\lceil \kappa_i \varepsilon^{-3/2} \rceil + 1)\kappa_j \right\rceil \\ & \leq \left\lceil \kappa_i \varepsilon^{-3/2} + 1 + (\kappa_i \varepsilon^{-3/2} + 2)\kappa_j \right\rceil \\ & = \left\lceil \varepsilon^{-3/2} \left(\kappa_i + \kappa_i \kappa_j + (1 + 2\kappa_j)\varepsilon^{3/2} \right) \right\rceil , \end{aligned} \tag{4.91}$$

where we used Lemma A.10 as well as the fact that $\lceil \kappa_i \varepsilon^{-3/2} \rceil \leq \kappa_i \varepsilon^{-3/2} + 1$ in step 3. Let us now note that $\varepsilon^{3/2} < 1$ and continue

$$\begin{aligned} |S_j| + |U_j| &\leq \left\lceil \varepsilon^{-3/2} (\kappa_i + \kappa_i \kappa_j + (1 + 2\kappa_j)) \right\rceil \\ &\leq \left\lceil (1 + \kappa_i)(2 + \kappa_j)\varepsilon^{-3/2} \right\rceil , \end{aligned} \tag{4.92}$$

which proves the assertion.

□

4.3.5 Discussion of Sampling Effects

The analysis of this Chapter shows that any sub-sampled cubic regularized method whose gradient and Hessian approximations satisfy 4.1 and 4.3 indeed retain the remarkable convergences properties of [Cartis et al., 2011] and [Nesterov and Polyak, 2006]. Specifically, sub-sampled methods that sample such that Theorem 4.8 and 4.10 hold retain these results with high probability (and regarding the local convergences: in expectation). Yet, it would be surprising if the cheaper SCR iterations came at no price at all. Let us thus take a closer look at the constants involved in the above derived convergence rates.

First off, Lemma 4.12 reveals that the supremum of $\{\sigma\}$ increases in the inaccuracy of the sub-sampled gradients and Hessians, i.e. ceteris paribus:

$$M \vee C \uparrow \Rightarrow \sigma_{\text{sup}} \uparrow . \tag{4.93}$$

This has the direct effect of *increasing* the number of unsuccessful steps SCR may take, which can be seen in Eq. (4.81).

Additionally, the upper bound σ_{sup} also appears in the denominator of (the bound on) κ_s in Eq. (4.49), where gradient and Hessian inaccuracies also show up. Thus we have, again ceteris paribus:

$$M \vee C \vee \sigma_{\text{sup}} \uparrow \Rightarrow \kappa_s \downarrow, \tag{4.94}$$

which in turn decreases the lower stepsize bound in Lemma 4.15. Thus, a lower value of κ_s has two negative effects. First, it *increases* the constant of the quadratic convergence result (Theorem 4.19 and secondly it *increases* the upper bound on the total number of successful steps that SCR may take via κ_i (see Eq. (4.89)).

As a result, the inaccuracy of the sub-sampled quantities may well lead to an increased overall number of SCR iterations. However, the total number of iterations stays within the same order of magnitude which suggests that the cheaper per-iteration cost overweight this effect. Empirical evidence for this claim is presented in the following Chapter.

4.4 Comparison with Trust Region Approaches

When viewing the penalty parameter σ_k as inversely proportional to the trust region radius Δ_k the algorithmic structure of cubic regularization and trust region methods is almost identical. Thus it is not surprising that similar convergence results can be found for trust region methods and comparable efforts are needed to solve the subproblems. To be specific, TR methods provide a global second-order convergence guarantee as well as a local quadratic convergence rate because eventually the trust region radius will be large enough to allow for full Newton steps.

Yet, the key difference between the two approaches is the effect that the regularization parameter updates have on the step size norm $\|s_k\|$, which can be seen in Figure 4.2 (see Section 3.2.2 for a derivation of the graphs).

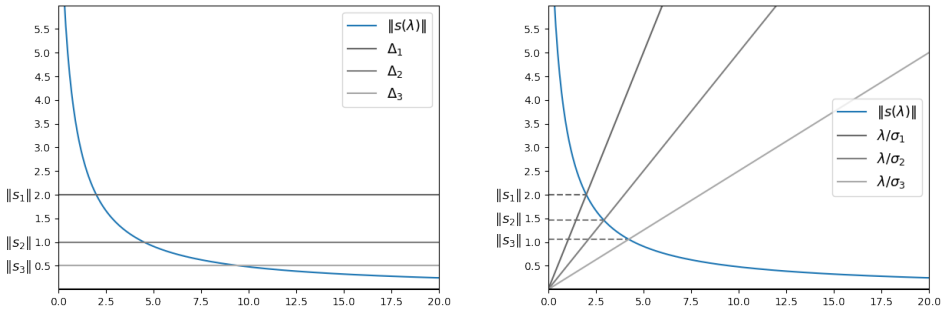


Figure 4.2: Step sizes of TR and ARC for different trust region radii $\Delta \in \{2, 1, 1/2\}$ and regularizes $\sigma \in \{1, 2, 4\}$ for $g = (1/4, 1/2)^\intercal$ and $H = diag(2, 1/2)$.

Even though both methods perform the same regularization updates, the

steps decrease over-proportionally in the trust region framework. Thus, the main drawback of trust region methods is that they may go from a large unsuccessful trial step to a successful but (too) short step. As a matter of fact, Curtis et al. [2014] show that the rate of decrease in $\|s\|$ is linear for trust region and only sublinear for cubic regularization methods[6].

To see this analytically note that if a trust region method never finds interior solutions, i.e. $\|s_k\| = \Delta_k, \forall k \geq 0$, we have (as stated in Section 2.5) that it behaves equivalently to a method that minimizes the quadratic model with a suitable *quadratic* regularizer $\hat{\sigma}_k$ in each iteration k

$$\hat{s}_k = \arg\min_{s \in \mathbb{R}^d} \hat{m}_k(s) := f(x_k) + s^\intercal g_k + \frac{1}{2} s^\intercal B_k s + \frac{\hat{\sigma}_k}{3} \|s\|^2 \tag{4.95}$$

and the model derivative changes to

$$\nabla \hat{m}_k(s) = g_k + B_k s + \hat{\sigma}_k s. \tag{4.96}$$

As a result the term $\sigma_k \|s_k\|^2$ in Eq. (4.41) of Lemma 4.14 becomes $\hat{\sigma}_k \|\hat{s}_k\|$ and we cannot establish the lower step size bound of Lemma 4.15 that relates $\|s_k\|$ to $\sqrt{\nabla f(w_{k+1})}$, which is one of the two crucial ingredients for the $O(\varepsilon^{-3/2})$ worst case complexity of cubically regularized methods. More detailes along with a rigorous proof of the $O(\varepsilon^{-2})$ worst case bound that holds for trust region methods can be found in [Gratton et al., 2008].

Recently, Curtis et al. [2014] designed a trust region variant with more sophisticated penalty parameter updates and step acceptance criteria to overcome precisely the above mentioned issue. They were indeed able to show that their method can achieve the $O(\varepsilon^{-3/2})$ first order worst case complexity but its practical relevance is questionable because of the increased complexity in the framework.

[6] for repeatedly unsuccessful iterations

Chapter 5

Experimental Results

In this section we present experimental results on synthetic as well as real-world data which largely confirm the analysis derived in the previous sections. The goal of our experiments is to investigate the performance of SCR in different problem settings. To be precise, we want to investigate the effect of

a) the number of datapoints n

b) the dimensionality d

c) the condition number κ

d) the presence and absence of convexity

on the performance of our method as well as baseline methods typically used in machine learning.

To this end, we first present results of logistic regressions on various real-world datasets. These are of fairly small dimensions but cover a wide range of problem sizes and condition numbers. In addition, the logistic loss function is per se convex but can be made non-convex by adding a suitable regularizer. Thus the first set of results covers the topics a), c) and d).

Subsequently, to be able to study the effect of an increasing dimensionality we generate three artificial datasets where the data itself arises from a multinominal Gaussian distribution. Again, we minimize a logistic loss function.

Finally, to foreshadow the applicability of sub-sampled cubic regularization methods in the context of learning neural networks we also present two results on image classification tasks that we train with multinominal logistic regression.

5.1 Datasets

Real-world Datasets

The real-world datasets we use represent very common instances of Machine Learning problems and are part of the libsvm library [Chang and Lin, 2011], except for *cifar* which is from Krizhevsky and Hinton [2009]. A summary of their main characteristic can be found in Table 5.1.

	type	n	d	$\kappa(H*)$	λ
a9a	Classification	32,561	123	761.8	$1e^{-3}$
a9a nc	Classification	32,561	123	1,946.3	$1e^{-3}$
covtype	Classification	581,012	54	$3 \cdot 10^9$	$1e^{-3}$
covtype nc	Classification	581,012	54	25,572,903.1	$1e^{-3}$
higgs	Classification	11,000,000	28	1,412.0	$1e^{-4}$
higgs nc	Classification	11,000,000	28	2,667.7	$1e^{-4}$
mnist	Multiclass	60,000	7,840	10,281,848	$1e^{-3}$
cifar	Multiclass	50,000	10,240	$1 \cdot 10^9$	$1e^{-3}$

Table 5.1: Overview over the real-world datasets used in our experiments with convex and non-convex (nc) regularizer

The multiclass datasets are both instances of so-called image recognition problems. The *mnist* images are greyscale and of size 28×28. The original *cifar* images are $32 \times 32 \times 3$ but we converted them to greyscale so that the problem dimensionality is comparable to *mnist*. Both datasets have 10 different classes, which multiplies the problem dimensionality by 10, giving the values in Table 5.1.

Synthetic Datasets

To test the influence of the dimensionality on the progress of the above applied methods we created artificial datasets of three different sizes, labeled as *gaussian s*, *gaussian m* and *gaussian l*.

	type	n	d	$\kappa(H^*)$	λ
gaussian s	Classification	50,000	100	2,083.3	$1e^{-3}$
gaussian m	Classification	50,000	1,000	98,298.9	$1e^{-3}$
gaussian l	Classification	50,000	10,000	1,167,211.3	$1e^{-3}$

Table 5.2: Overview over the synthetic datasets used in our experiments with convex regularizer

The feature vectors $X = (x_1, x_2, ..., x_d), x_i \in \mathbb{R}^n$ were drawn from a multivari-

ate Gaussian distribution

$$X \sim \mathcal{N}(\mu, \Sigma) \tag{5.1}$$

with a mean of zero $\mu = (0, \ldots, 0)$ and a covariance matrix that has reasonably uniformly distributed off-diagonal elements in the interval $(-1, 1)$.

5.2 Implementations

5.2.1 Practical Implementation of SCR

We implement SCR as stated in Algorithm 1 and note the following details. Following Erdogdu and Montanari [2015], we require the sampling conditions derived in Section 4.2 to hold with probability $O(1 - 1/d)$, which yields the following practically (almost) applicable sampling schemes

$$\begin{aligned} |\mathcal{S}_{B,k}| &\geq \frac{36\kappa_g^2 \log(d)}{(C\|s_k\|)^2},\ C > 0,\ \forall k > 0 \\ |\mathcal{S}_{g,k}| &\geq \frac{32\kappa_f^2(\log(d) + 1/4)}{M^2\|s_k\|^4},\ M > 0,\ \forall k > 0. \end{aligned} \tag{5.2}$$

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 the κ_g and κ_f, which are generally expensive to obtain. One way to estimate κ_g would be using the quantity $(B_{k-1}s_{k-1})/\|s_{k-1}\|)$ but empirically this made no significant (positive or negative) difference on our datasets.

As argued in Section 3.3 the use of stochastic gradient estimates is not very likely to improve the performance of SCR as long as n stays below the point where evaluating full gradients becomes overwhelmingly costly[1]. Preliminary experiments confirm this intuition[2]. We thus make use of full first-order information and only sub-sample the Hessian according to Eq. (5.2) in the following.

However, when choosing $|\mathcal{S}_{B,k}|$ for the current iteration k, the stepsize 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 and this is confirmed by the experimental results. Finally we point out that the sampling schemes derived in Eq. (5.2) 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 (see Section 4.2.3).

[1] Note that this threshold would certainly decrease if we had way of solving the subproblems linearly or even logarithmically in d.

[2] In fact, on *gaussian s* the first n that made gradient sampling competitive was $n = 22,000,000$

The regularization parameter updating is analog to the rule used in the reported experiments of Cartis et al. [2011], where $\gamma = 2$. The goal is to reduce the regularization penalty rapidly as soon as convergence sets in, while keeping some regularization in the non asymptotic regime. Regarding the successfulness thresholds we chose $\eta_1 = 0.2$, and $\eta_2 = 0.8$. As initial (and minimum) sample size we chose a value between 2.5% and 10%, depending on the problem.

5.2.2 Other Methods

We here briefly describe the choice of hyper-parameters for the baseline algorithms. Further details on these methods can be found in Section 2.3.

- Stochastic Gradient Descent (SGD): To bring in some variation, we select a mini-batch of the size $\lceil n/10 \rceil$ on the real world classification- and $\lceil n/100 \rceil$ on the multiclass problems. On the artificial datasets we only sample 1 datapoint per iteration and update the parameters with respect to this point. We use a problem-dependent, constant step-size as this yields faster initial convergence [Hofmann et al., 2015],[Roux et al., 2012].
- SAGA: is a variance-reduced variant of SGD that only samples 1 datapoint per iteration and uses a constant step-size.
- Broyden-Fletcher-Goldfarb-Shanno (BFGS) is the most popular and stable Quasi-Newton method.
- Limited-memory BFGS is a variant of BFGS which uses only the recent K iterates and gradients to construct an approximate Hessian. We used $K = 20$ in our experiments. Both methods employs a line-search technique that satisfies the strong Wolfe condition to select the step size.
- NM is the classic version of Newton's method which we apply with a backtracking line search.

For L-BFGS and BFGS we used the implementation available in the optimization library of scipy. All other methods are our own implementation. During this thesis we developed in particular: a) the trust region framework with various subproblem solvers, b) the adaptive cubic regularization framework with various subproblem solvers, c) the bi- as well as multinominal logistic loss function as well as their derivatives and d) the non-convex regularizer as well as its derivatives.

The source code is available on github. It is optimized with respect to runtime, i.e. all algebraic operations are coded as vectorized operations in *numpy* which we build against Intels high performance MathKernelLibrary. For the multinominal logistic loss we apply an efficient, backpropagation

like approach for evaluating Hessian-vector product in $O(nd)$ that is due to Pearlmutter [1994].

All experiments with $d < 1000$ were run on a machine with 8 GB memory and a CPU with a 2.4 GHz nominal clock rate. The others were run on a machine with 32 GB memory and 2.8 GHz nominal CPU clock rate.

5.3 Results

The following results largely confirm our analysis since **SCR** reduces the runtime of ARC on all instances without losing its global convergence property. The deterministic ARC method performs consistently well. While it is mostly comparable to TR in terms of runtime, it has a little edge in terms of epochs[3].

In general, none of the **first-order methods** provides comparable (log) suboptimality values in reasonable time. Both methods perform significantly worse on problems with higher condition numbers. As expected SAGA generally outperforms SGD in terms of runtime and samples seen. The latter quickly enters the so-called zig-zagging behaviour which can best be observed in Figure 5.2 and 5.3, where lower batch sizes were used. SAGA is not in the *higgs* plot since creating the stored gradient table itself takes longer than most other methods needs to converge.[4]

The **quasi-Newton algorithms** perform somewhere in between the first and second-order methods, which reflects their hybrid nature. Interestingly, Nocedal and Wright [2006] state that the main weakness of L-BFGS is its slow convergence on ill-conditioned problems (without giving any analytical reason). Our results clearly support this claim since L-BFGS performs comparably bad on *covtype* and *cifar*.

5.3.1 Influence of Size, Conditioning and Convexity

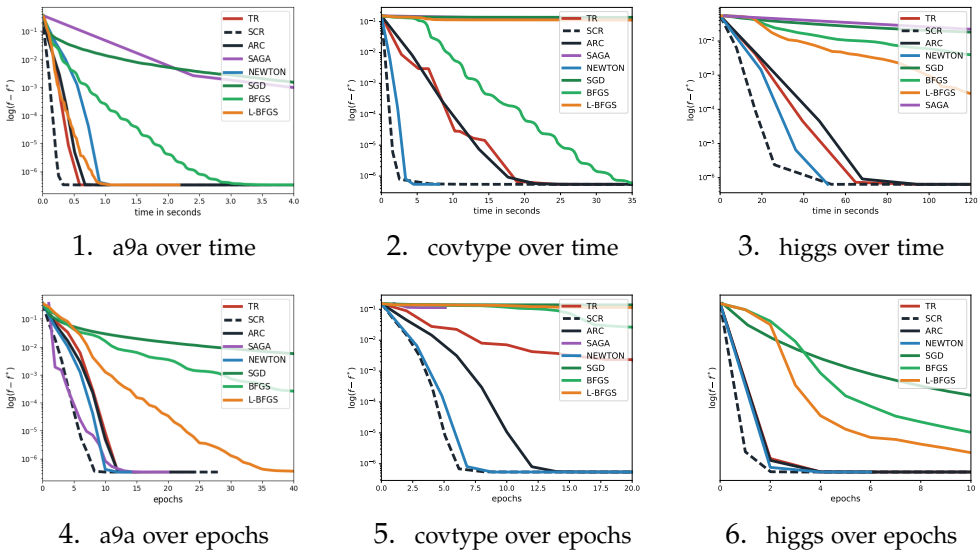
Convex Problems

For the first set of problems we added the classical ℓ_2 regularization $\lambda\|w\|^2$ to the objectives to make them strongly convex. Figure 5.1 shows the results. Compare *a9a* and *higgs* for the in the influence of n since they have very similar characteristics. The influence of the conditioning can be studied when comparing the ill-conditioned *higgs* dataset to the other two.

Since all of the problems in this subsection are fairly low dimensional, Newton's method does quite well. Especially on *covtype* where second-order

[3] Remember that the theoretical superiority of cubic regularization methods is indeed in terms of iterations and not runtime.

[4] Furthermore, the memory requirement sums up to about 2500 megabyte. The creation of this table is why SAGA starts one epoch to the right of all others in the Figures.



1. a9a over time 2. covtype over time 3. higgs over time

4. a9a over epochs 5. covtype over epochs 6. higgs over epochs

Figure 5.1: Log suboptimality of the empirical risk averaged over 10 independent runs on convex problems

information seems to be crucial[5]. In this case ARC and TR take more time to optimize but are in a comparable range of epochs which suggests that the additional overhead due to the more complex framework plays a role in the runtime.

Non-convex Problems

Figure 5.2 shows the performance of the above mentioned methods on the same problems with the following non-convex regularizer

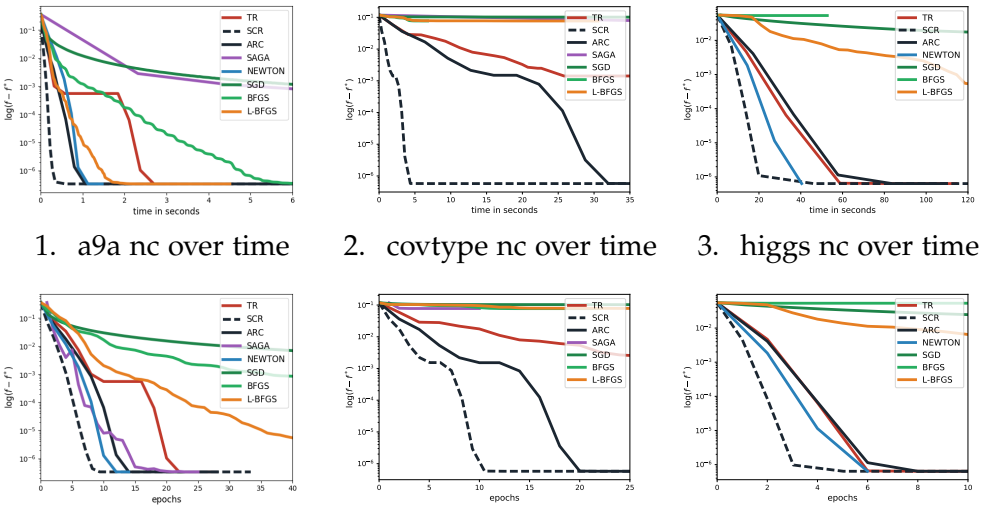
$$r(w) = \sum_i^d \frac{w_i^2}{1+w_i^2}, \quad \nabla r(w) = \begin{pmatrix} \frac{2w_1}{(w_1^2+1)^2} \\ \vdots \\ \frac{2w_d}{(w_d^2+1)^2} \end{pmatrix}, \quad \nabla^2 r(w) = diag\left(\begin{pmatrix} \frac{2-6w_1^2}{(w_1^2+1)^3} \\ \vdots \\ \frac{2-6w_d^2}{(w_d^2+1)^3} \end{pmatrix} \right), \tag{5.3}$$

which we applied to investigate the second-order convergence results of SRC. The function $r(w)$ is obviously non-convex as soon as some $w_i > \sqrt{1/3}$. Yet, as expected, the presence of saddle points does not prevent SCR and ARC from finding a global minimizer in reasonable time.

As a matter of fact, all methods converge on *a9a* but on *covtype* and *higgs* they suffer clearly from the presence of non-convexity. Newton's method

[5] compare the condition numbers in Table 5.1)

cannot optimize *covtype* because at some iterate it encounters a singular Hessian. Interestingly, on *higgs*, BFGS steps to a saddle point and terminates the optimization process.



1. a9a nc over time 2. covtype nc over time 3. higgs nc over time

5. a9a nc over epochs 6. covtype nc over epochs 7. higgs nc over epochs

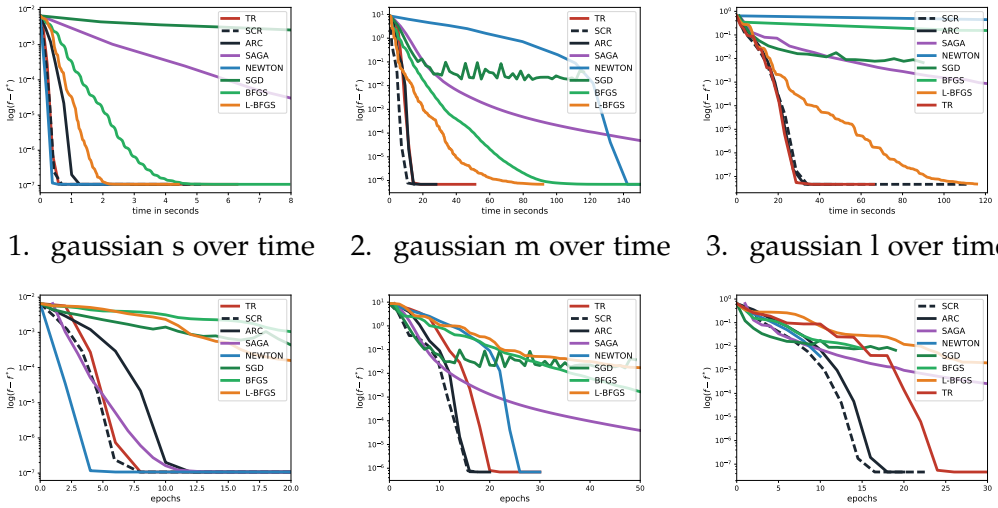
Figure 5.2: Log suboptimality of the empirical risk averaged over 10 independent runs on non-convex problem

5.3.2 Influence of Dimensionality

As expected Newton's method is slowed down severely by the increase in d since the factorization of the Hessian rises cubically in its dimension. Consider Figure 5.3. On the *gaussian l* dataset it needs 27.8 minutes to find the optimum whereas the hessian-free approaches (TR, ARC and SCR) reach approximate optimality in a matter of seconds. They scale comparably very well since they only need indirect access to the Hessian via matrix-vector products. Evidently, these methods outperform also the quasi-newton approaches even in high dimensions. Among these, the limited memory version of BFGS is significantly faster than original variant.

5.3.3 Multiclass Problems

In this section we leave the trust region method out because our implementation is not optimized towards solving multi-class problems. We do not run Newton's method or BFGS either as the above results suggests that they are unlikely to be competitive. Furthermore, Figure 5.4 does not show loga-



1. gaussian s over time 2. gaussian m over time 3. gaussian l over time

5. gaussian s over epochs 6. gaussian m over epochs 7. gaussian l over epochs

Figure 5.3: Log suboptimality of the empirical risk averaged over 10 independent runs on convex-problems

rithmic but linear suboptimality, because optimizing these problems to high precision takes very long and yields few additional benefits[6]. The interested reader can find long term results in the Appendix (Figure A.1).

As a result these kind of problems are hardly ever optimized to high precision. Instead, "early stopping" is applied which is reasonable to do since the marginal utility of the last digits of precision is fairly low because after all what we minimize is the *expected risk* but our original goal is for the method to generalize well to unseen data[7].

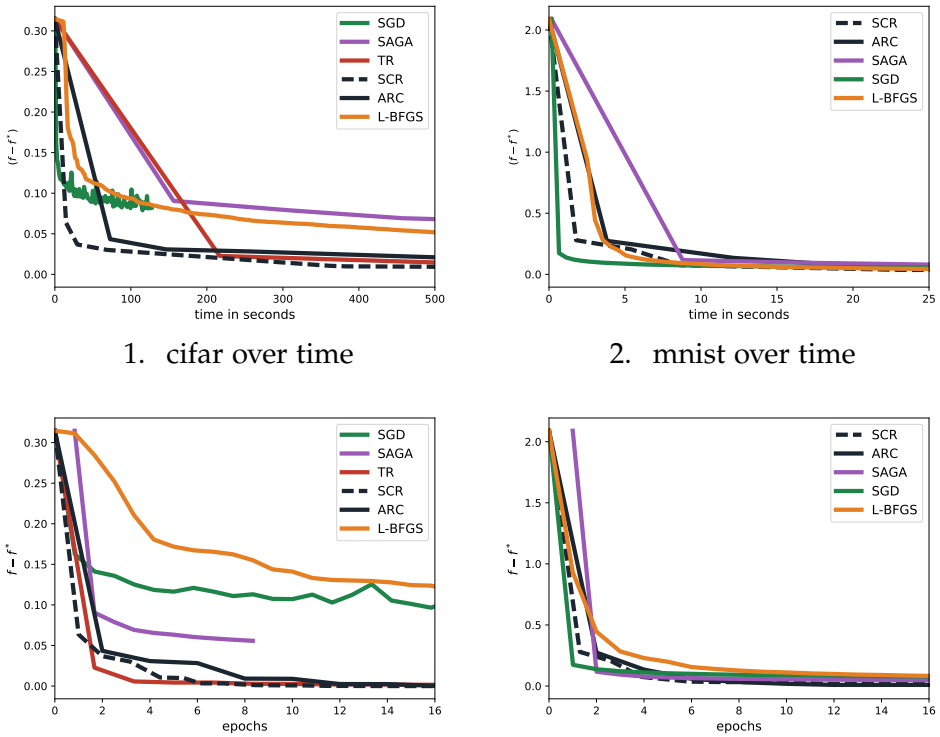
In this regard, SCR yields early progress at a comparable rate to other methods but gives the opportunity to solve the problem to high precision if needed.

5.3.4 Conclusion

Unsurprisingly, the runtime of all algorithms increases in **n**. Yet, the second-order methods (ARC, SCR, TR and Newton's method) suffer comparably less and n has no influence in terms of epochs.

[6]For example, the 25th SCR iteartion drove the gradient norm from $3.8 \cdot 10^{-5}$ to $5.6 \cdot 10^{-8}$ after building up a Krylov space of dimensionality 7800. It took 9.47 hours and did *not* change any of the first 13 digits of the loss.

[7]In the field of machine learning this is sometimes seen as a type of regularization to prevent overfitting



1. cifar over time
2. mnist over time
3. cifar over epochs
4. mnist over epochs

Figure 5.4: Suboptimality of the empirical risk

A higher dimensionality **d**, in turn, increases both runtime and iterations of especially the second-order and quasi-newton methods. The effect goes so far that Newton's method and BFGS are hardly applicable in dimensions of 10,000 or higher. However, the Hessian free variants scale comparably well. SCR and ARC perform more and more similar as d increases which is due to the fact that the our sampling scheme (Eq. (5.2)) tends to give sample sizes close to n already after a few iteration in such high dimensions.[8]

On the other hand, the results from *covtype* and *cifar* suggest that sampling may be particularly efficient on ill-conditioned problems, i.e. when κ is high. While ill-conditioning is generally perceived as an obstacle for optimization algorithms, there is indeed a rational behind this effect. In our setting, one source of a high eigenvalue spectrum of the Hessian is the presence of correlation in the dataset X. For example, the Hessian of a linear regression model is $X^\intercal X$. A closer look at this matrix reveals that its diagonal con-

[8]Preliminary experiments with lower sample sizes (exponential increase) yielded *longer* runtimes than ARC and SCR.

tains variance terms and the off-diagonal elements are the covariances of the columns of X. Furthermore, it is intuitive that the higher the covariance and thus redundancy in the data, the more meaningful it becomes to sub-sample datapoints. Since most datasets of the era of ”Big Data” presumably contain at least some redundancy, this is an argument for the use of SCR in machine learning.

Finally, the introduction of **non-convexity** appears to have a negative effect on both, runtime and epochs. The combination of non-convexity and ill-conditioning, as it can be found on *covtype*, yields particularly bad results. Here, SCR provides the greatest savings which shows that even when second-order information is clearly important, carefully chosen sub-samples may provide sufficient information at comparably low cost. This reinforces the above given argument for the use of sub-sampled cubic regularization methods also in non-convex settings.

Chapter 6

Conclusion and Future Work

Conclusion This thesis laid out the foundational work for a class of stochastic cubic regularization methods that make use of sub-sampled gradient and Hessian estimates for solving large-scale non-convex learning problems.

Specifically, we extended the adaptive cubic regularization approach of Cartis et al. [2011] to a fully stochastic framework that allows the use of inexact gradients and Hessians. Using concentration inequalities we developed sampling conditions that are sufficient to retain the convergence properties of the deterministic method, which include the best known global convergence rate on non-convex functions. To the best of our knowledge, the Hessian sampling scheme we propose is the first Hessian approximation technique that provably yields sufficient second-order information for these guarantees to hold. Furthermore, this is the first work to explore sub-sampled cubic regularization methods for applications in machine learning.

Numerical experiments on both, real and synthetic datasets demonstrate the superior performance of the proposed algorithm over its deterministic counterpart. In addition, the general framework of cubic regularization methods has proven to be preferable to the use of stochastic gradient descent and quasi-Newton methods over a wide range problems.

Future work Perhaps the most exciting direction for future research is the application of sub-sampled cubic regularization methods in the context of Deep Learning. Based on our theoretical analysis these methods are well-equiped to tackle this problem and the pioneering work of Martens [2010] suggests their practicability. Towards this end, a major improvement would be to reduce the computational complexity of solving the subproblems in very high dimensions.

Fortunately, the issue of solving trust region and cubic regularization subproblems has been under continuous investigation during the past two decades.

Remarkable recent results include the work of Agarwal et al. [2016a] and Carmon and Duchi [2016] as well as Carmon et al. [2016]. The former minimize the subproblems approximately in $\mathbb{R}^d$ using a safeguarded binary search that makes only logarithmic number of guesses on λ^*. Thus, they achieve an overall runtime that is linear in d. Carmon and Duchi [2016] show that a sophisticated version of GD is sufficient to approximate the *global* model minimizer regardless of the multiple saddle points and local minima of the objective. The convergence rate scales only logarithmically in the dimension. However, they need to find a good initial point for GD in the first place which requires two Hessian-vector products. Furthermore, both approaches loose the $O(\varepsilon^{-3/2})$ worst case guarantee. Specifically, the first approach yields an $O(\varepsilon^{-7/4})$ rate while the second only provide a $O(\varepsilon^{-2})$ worst case bound.

Another interesting direction to follow is the field of importance sampling techniques that create a non-uniform sampling distribution $\{p_i\}_{i=1}^n$ to capture the varying amounts of information contained by each datapoint in each iteration. Interestingly, spectral approximation techniques such as leverage score based (online) row sampling [Cohen et al., 2016] can be extended to sampling datapoints. For example, Xu et al. [2016] achieve a linear-quadratic local convergence rate by applying two non-uniform sub-sampled newton methods. The first samples based on the block norm squares, which is basically a Frobenius norm ratio and the second on (block partial) leverage score.

For problems with such high numbers of datapoints that make evaluating the full gradient too costly, sub-sampling gradients according to our framework could be combined with forming coarse Hessian approximations, e.g. as diagonal matrix or via sketching techniques as studied by Pilanci and Wainwright [2016]. As a matter of fact lower dimensional hessian approximations would be beneficial regarding the runtime because of their quadratic presence in the subproblem solver complexity.

Regarding the outer cubic regularization framework, it should both possible and beneficial to decouple the penalty parameter updates from the inexactness of the stochastic quantities. For this purpose, sophisticated successfulness measures need to be designed. These may likely offset some of the increase in the convergence constants that we attributed to the inexact first- and second-order information in Chapter 4.

In summary, such developments would lead to new, light-weight randomized cubic regularization methods that scale well to high dimensional problems such as deep learning.

Bibliography

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

Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. *arXiv preprint arXiv:1602.03943*, 2016b.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. *arXiv preprint arXiv:1603.05643*, 2016.

Anima Anandkumar and Rong Ge. Efficient approaches for escaping higher order saddle points in non-convex optimization. *arXiv preprint arXiv:1602.05908*, 2016.

Garrett Birkhoff and Saunders Mac Lane. *A survey of modern algebra*. Universities Press, 1966.

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.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Alan J Bray and David S Dean. Statistics of critical points of gaussian fields on large-dimensional spaces. *Physical review letters*, 98(15):150201, 2007.

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

Yair Carmon and John C Duchi. Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv preprint arXiv:1612.00547*, 2016.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *arXiv preprint arXiv:1611.00756*, 2016.

Coralia Cartis, Nicholas IM Gould, and Ph L Toint. On the complexity of steepest descent, newton's and regularized newton's methods for nonconvex unconstrained optimization problems. *Siam journal on optimization*, 20 (6):2833–2852, 2010.

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

Coralia Cartis, Nicholas IM Gould, and Philippe L Toint. *How Much Patience to You Have?: A Worst-case Perspective on Smooth Noncovex Optimization*. Science and Technology Facilities Council Swindon, 2012.

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

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

Michael B Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *arXiv preprint arXiv:1604.05448*, 2016.

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

Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of\ mathcal {O}(\ epsilonˆ{-3/2}) for nonconvex optimization. *Mathematical Programming*, pages 1–32, 2014.

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

John E Dennis and Jorge J Moré. A characterization of superlinear convergence and its application to quasi-newton methods. *Mathematics of computation*, 28(126):549–560, 1974.

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

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

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Nicholas IM Gould, Stefano Lucidi, Massimo Roma, and Philippe L Toint. Solving the trust-region subproblem using the lanczos method. *SIAM Journal on Optimization*, 9(2):504–525, 1999.

Robert Mansel Gower. *Sketch and Project: Randomized Iterative Methods for Linear Systems and Inverting Matrices*. PhD thesis, The University of Edinburgh, 2016.

Serge Gratton, Annick Sartenaer, and Philippe L Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

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

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

Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313, 2015.

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

Hubertus Th Jongen, Peter Jonker, and Frank Twilt. *Nonlinear optimization in finite dimensions: Morse theory, Chebyshev approximation, transversality, flows, parametric aspects*, volume 47. Springer Science & Business Media, 2013.

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

Richard Kueng and David Gross. Ripless compressed sensing from anisotropic measurements. *Linear Algebra and its Applications*, 441:110–123, 2014.

Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic newton method. *arXiv preprint arXiv:1503.08316*, 2015.

James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

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

Jorge Nocedal and Stephen J Wright. Numerical optimization, second edition. *Numerical optimization*, pages 497–528, 2006.

Victor Y Pan and Zhao Q Chen. The complexity of the matrix eigenproblem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 507–516. ACM, 1999.

Ioannis Panageas and Georgios Piliouras. Gradient descent converges to minimizers: The case of non-isolated critical points. *arXiv preprint arXiv:1605.00405*, 2016.

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

Mert Pilanci and Martin J Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, 17(53):1–38, 2016.

Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods i: globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016a.

Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods ii: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016b.

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

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

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *heory of Probability and Its Applications*, 16(2):264–280, 1971.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

Appendix A

Appendix

Feasibility of subproblems

Lemma A.1 (Weierstrass extreme value theorem) *If M is a compact set in $\mathbb{R}^d$, and f a continuous function from M to $\mathbb{R}$, then f has a global maximum and a global minimum on M.*

See [Birkhoff and Mac Lane, 1966] for a sketch of the proof.

Lemma A.2 (Coercivity and compactness) *Let $f \in C^2(\mathbb{R}^d, \mathbb{R})$. If f is furthermore coercive then for every $\alpha \in \mathbb{R}$ the level set $\text{lev}_{\leq}^{\alpha} = \{w | f(w) \leq \alpha\}$ is compact.*

Proof: Obviously, the continuity of f implies that the sets $\text{lev}_{\leq}^{\alpha}$ are closed. To show that they are furthermore bounded, suppose $\exists \hat{alpha} \in \mathbb{R}$ such that $\text{lev}_{\leq}^{\hat{alpha}}$ is unbounded. Then there must exist a sequence $\{w^\nu\} \in \text{lev}_{\leq}^{\hat{alpha}}$ with $\lim_\nu \|w^\nu\| = +\infty$ which by the coercivity of f implies $f(w^\nu) \to \infty$. However, this is a clear contradiction to $f(w^\nu) \leq \alpha, \forall \nu = 1, 2, ...$ and thus $\text{lev}_{\leq}^{\hat{alpha}}$ must be bounded.

□

Note that the above lemma actually holds with *if and only if* but for the sake of brevity we only present the direction that is relevant for our analysis.

Convergence of random sequences

The method presented here depends on random quantities, in particular random gradients and random Hessian matrices in each iteration. Hence, the iterates generated by the random models constitute a sequence of random variables themselves. We shall thus establish the following two concepts of convergence of a random sequence.

Definition A.3 (Convergence of a random sequence) *Consider a sequence of random variables $\{w_k\}$. Firstly, we say that the **norm of the expectation** of $\{w_k\}$ converges to w_* with rate $c \in [0, 1)$, if*

$$\|\mathbb{E}[w_k] - w_*\| \leq c^k \|w_0 - w_*\|, \tag{A.1}$$

which is equivalent to

$$\frac{\|\mathbb{E}[w_{k+1} - w_*]\|}{\|w_k - w_*\|} \leq c \tag{A.2}$$

*Furthermore, this implies that $\mathbb{E}[w_k] \to 0$ and thus the sequence **converges in expectation**.*

*Second, we say that the **expected norm** of $\{w_k\}$ converges to w_* with rate $c \in [0, 1)$, if*

$$\mathbb{E}\left[\|w_k - w_*\|^2\right] \leq c^k \|w_0 - w_*\|^2 \tag{A.3}$$

which is equivalent to

$$\frac{\mathbb{E}\left[\|w_{k+1} - w_*\|\right]^2}{\|w_k - w_*\|^2} \leq c \tag{A.4}$$

*This implies both, convergence in expectation and **convergence in probability** (Gower [2016]), which is characterized as follows. For any $\varepsilon > 0$*

$$\mathbb{P}(\|w_k - w_*\|^2 \geq \varepsilon \|w_0 - w_*\|^2) \to 0, \text{as } k \to \infty. \tag{A.5}$$

Superlinear convergence

We now derive a superlinear convergence result that can be obtained *without* assuming a lipschitz continous Hessian and furthermore by only requiring the Dennis More condition for the Hessian agreement *and* a similarly weaker gradient agreement condition which we shall state first.

Assumption A.4 (Weak Agreement of ∇f and g^s)

$$\frac{\|\nabla f(w_k) - g^s(w_k)\|}{\|s_k\|} \to 0, \text{ whenever } \|g_k\| \to 0. \tag{A.6}$$

Theorem A.5 (Superlinear local convergence in expectation) *Let $f \in C^2$, ∇f Lipschitz continuous and B_k bounded above. Let s_k satisfy (3.53) and*

$$w_k \to w_*, \text{ as } k \to \infty, \tag{A.7}$$

where $H(w_*)$ is positive definite. Moreover, assume that TC is satisfied with $g_k \to 0, k \to \infty, k \in \mathcal{S}$. Then, if $\mathbb{E}\left[g_k^s\right] = \nabla f_k$ and $\mathbb{E}\left[B_k\right] = H_k$

$$\frac{\|\mathbb{E}\left[\nabla f(w_{k+1})|w_k\right]\|}{\|\nabla f(w_k)\|} \leq c_k,\ c_k \to 0,\ \text{as } k \to \infty \tag{A.8}$$

and

$$\frac{\|\mathbb{E}\left[w_{k+1} - w_*|w_k\right]\|}{\|w_k - w_*\|} \leq c_k,\ c_k \to 0,\ \text{as } k \to \infty. \tag{A.9}$$

That is, w_k converges to w_*, superlinearly as $k \to \infty$.

Proof: TC and Asm.3.7 (thus (3.53)) are defined for any possible model minimizer s_k and thus also apply for the expected value of s_k. Consequently, the deterministic analysis carries over to the stochastic case and it is most convenient to directly apply $\mathbb{E}\left[g_k^s\right] = \nabla f_k \to \mathbb{E}\left[e_k\right] = 0$, $\mathbb{E}\left[B_k\right] = H_k$ and $\mathbb{E}\left[s_k\right]$ at the very end of the original proof.

$$\begin{aligned}\frac{\|\mathbb{E}\left[\nabla f(w_{k+1})\right]\|}{\|\nabla f(w_k)\|} \leq& \left(1 + \frac{\|\mathbb{E}\left[e_k\right]\|}{\|\nabla f(w_k)\|}\right)\left(\frac{R_{\min}\mathbb{E}\left[d_k\right] + \sigma_{\sup}(\|\nabla f(w_k)\| + \|\mathbb{E}\left[e_k\right]\|)}{R_{\min}^2(1-\kappa_\theta)}\right)\\ =& \frac{R_{\min}\mathbb{E}\left[d_k\right] + \sigma_{\sup}\|\nabla f(w_k)\|}{R_{\min}^2(1-\kappa_\theta)},\end{aligned} \tag{A.10}$$

where

$$\begin{aligned}\mathbb{E}\left[d_k\right] =& \|\int_0^1 (H(w_k + t\mathbb{E}\left[s_k\right]) - H(w_k))dt\| + \frac{\|(H(w_k) - \mathbb{E}\left[B_k\right])s_k\|}{\|\mathbb{E}\left[s_k\right]\|} + \kappa_\theta\kappa_g\mathbb{E}\left[g_k\right]\\ &+ (1 + \kappa_\theta\kappa_g)\frac{\|\mathbb{E}\left[e_k\right]\|}{\|\mathbb{E}\left[s_k\right]\|}\\ =& \|\int_0^1 (H(w_k + t\mathbb{E}\left[s_k\right]) - H(w_k))dt\| + \kappa_\theta\kappa_g\mathbb{E}\left[g_k\right]\end{aligned} \tag{A.11}$$

Since, asymptotically, the first summand goes to zero and $g_k \to 0$ per assumption, also $\mathbb{E}\left[g_k\right] \to 0$. Together, this implies $\mathbb{E}\left[d_k\right] \to 0$, as $k \to \infty$. Additionally, we have that $\nabla f(w_k) \to 0$ when approximating w^* from Theorem (4.20) and we can finally note that

$$\frac{\|\mathbb{E}\left[\nabla f(w_{k+1})\right]\|}{\|\nabla f(w_k)\|} \leq c_k \text{ where } \lim_{k\to\infty} c_k \to 0. \tag{A.12}$$

Consequently, the sequence of iterates generated by sARC converges superlinearly in expectation.

□

Equivalently to the quadratic convergence case the use of full gradients makes taking the expected value obsolete.

Convergence of gradients and iterates

Lemma A.6 *Let $f \in C^2$, the Hessian bounded above and σ_k bounded above and below (4.12). Furthermore let $\{w_k\} \to w^*$, with w^* 2nd-order critical point. Then*

$$\frac{\|g_{k+1}\|}{\|g_k\|} \to 0 \text{ also } \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|} \to 0 \tag{A.13}$$

Proof: Let us first introduce a multidimensional version of Taylor's Theorem.

Definition A.7 (Multidimensional Taylor Theorem (1-jet)) *Let $g : \mathbb{R}^n \to \mathbb{R}^m$ and $g \in C^2$ then:*

$$g(y) = g(x) + D(g(x))(y - x) + O(\|y - x\|^2) \tag{A.14}$$

with $D(g(w)) \in \mathbb{R}^{n \times n}$ the Jacobi-Matrix of g.

Since the Hessian of f equals the transpose of this Jacobi-Matrix and is furthermore symmetric for $f \in C^2$ we have $D(g(w)) = H(w)$. Thus follows by Taylor expansions of g_k and g_{k+1} around w^* that we have

$$\|g_k\| = \|g(w^*) + H(w^*)(w_k - w_*) + O(\|w_k - w^*\|^2)\|,$$

where $O(\|w - w^*\|^2)$ is of the form $\omega(w)\|w - w^*\|^2$ with $\lim_{w \to w^*} \omega(w) = \omega(w^*) = 0$. Thus for $w_k \to w^*$ and $g(w^*) = 0$ as well as $H(w^*) \succ 0$ and $\|H\| \leq \kappa_H$:

$$\frac{\|g_{k+1}\|}{\|g_k\|} = \frac{\|H(w^*)(w_{k+1} - w^*)\|}{\|H(w^*)(w_k - w^*)\|} \geq \frac{\sigma_{\min}(H(w^*))\|(w_{k+1} - w^*)\|}{\sigma_{\max}(H(w^*))\|w_k - w^*\|} \tag{A.15}$$

Finally, since $\|\cdot\| \geq 0$ and by the above assumptions $0 < \sigma_{min} < \sigma_{max} \leq \kappa_H$ we get:

$$\frac{\|g_{k+1}\|}{\|g_k\|} \to 0 \text{ also } \frac{\|w_{k+1} - w^*\|}{\|w_k - w^*\|} \to 0 \tag{A.16}$$

□

Comment on Eq (A.15):

From a SVD of the Hessian we have: $Hv = U\Sigma V^T v$, with U and V orthogonal and since orthogonal transformations preserve length, i.e. $\|Hv\| = \|v\|$ we have:

$$\|Hv\|_2 = \|U\Sigma V^T v\|_2 = \|U(\Sigma V^T)v\|_2 = \|V^T(\Sigma v)\|_2 \tag{A.17}$$

$$= \|\Sigma v\|_2 = \sqrt{\sigma_1^2 v_1^2 + ... + \sigma_n^2 v_n^2} \tag{A.18}$$

$$\geq \sqrt{\sigma_{\min}^2(v_1^2 + ... + v_n^2)} = \sigma_{\min}(H)\|v\|_2 \tag{A.19}$$

For the denominator we apply the upper bound $\|Hv\| \leq \|H\|\|v\| \leq \sigma_{max}\|v\|$

Miscellaneous

Lemma A.8 *Let Q and A square matrices of the same size and furthermore Q orthogonal. Then the Eigenvalues λ of $Q^\intercal AQ$ and A coincide.*

Proof: Let $v \in \mathbb{R}^n \setminus \{0\}$ an eigenvector of A, i.e. $Av = \lambda v$. Then λ is an eigenvalue of $Q^\intercal AQ$ with eigenvector $Q^\intercal v$:

$$(Q^\intercal AQ)(Q^\intercal v) = Q'A(QQ^\intercal)v = Q^\intercal Av = \lambda(Q^\intercal v).$$

Let $v \in \mathcal{L} \setminus \{0\}$ an eigenvector of $(Q^\intercal AQ)$, i.e. $(Q^\intercal AQ)v = \lambda v$. Then λ is an eigenvalue of A with eigenvector Qv:

$$A(Qv) = IAQv = QQ^\intercal AQv = Q(Q^\intercal AQv) = \lambda(Qv).$$

In both cases we used the following characterization of orthogonal matrices $Q^\intercal Q = QQ^\intercal = I$, which only holds for square matrices.

□

Definition A.9 (Rayleigh Coefficient) *If a matrix A is symmetric and the vector $w \neq 0$ than we call the scalar*

$$R(w) \stackrel{def}{=} \frac{w^\intercal Bw}{\|w\|^2} \tag{A.20}$$

the Rayleigh coefficient of w.

This quotient has the nice property of lying between the left- and rightmost eigenvalues of A, that is

$$\lambda_{\min}(A) \leq R(w) \leq \lambda_{\max}(A). \tag{A.21}$$

Lemma A.10 (Upper bound on ceil of sum) *Let $\beta_1, \beta_2 \in \mathbb{R}_{\geq 0}$ then*

$$\lceil \beta_1 \rceil + \lceil \beta_2 \rceil \leq \lceil \beta_1 + \beta_2 \rceil + 1 \tag{A.22}$$

Proof: Let us denote by $\alpha_1, \alpha_2 \in [0, 1)$ the decimal digits of β_1 and β_2 respectively. Then we can rewrite every β_i as

$$\beta_i = \lfloor \beta_i \rfloor + \alpha_i$$

, $i \in \{1, 2\}$ and since $\lceil z + \beta \rceil = z + \lceil \beta \rceil$ for each $z \in \mathbb{Z}$, inequality (A.22) becomes

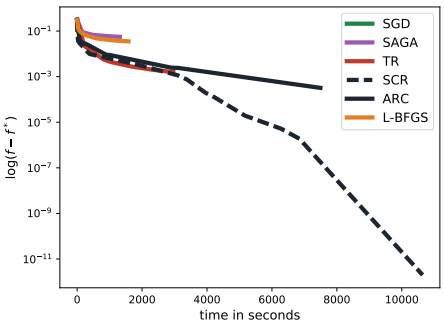
$$\lfloor \beta_1 \rfloor + \lfloor \beta_2 \rfloor + \lceil \alpha_1 \rceil + \lceil \alpha_2 \rceil \leq \lfloor \beta_1 \rfloor + \lfloor \beta_2 \rfloor + \lceil \alpha_1 + \alpha_2 \rceil + 1.$$

Hence, the crucial quantities for investigating the order in (A.22) are the decimal digits and its respective ceils. There are three cases to distinguish.

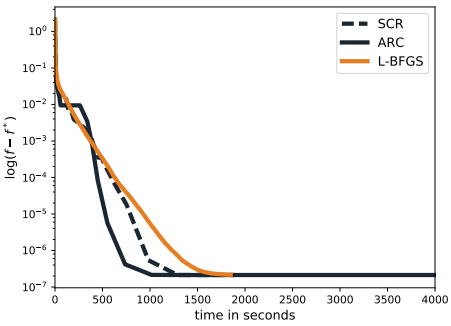
(a) If β_1 and β_2 are in $\mathbb{Z}$ then $\alpha_1 = \alpha_2 = \lceil \alpha_1 \rceil = \lceil \alpha_2 \rceil = \lceil \alpha_1 + \alpha_2 \rceil = 0$ and (A.22) holds since $0 \leq 1$. (b) If either one of β_1, β_2 is in $\mathbb{Z}$ and the other is not, we have for example $a_1 \in (0, 1)$ and $a_2 = 0$ so that $(a_1 + a_2) \in (0, 1)$ and thus $\lceil \alpha_1 \rceil + \lceil \alpha_2 \rceil = 1 \leq \lceil \alpha_1 + \alpha_2 \rceil + 1 = 2$. Finally, (c) if both $\beta_1, \beta_2 \notin \mathbb{Z}$ we have $\alpha_1, \alpha_2 \in (0, 1)$ and thus $\lceil \alpha_1 \rceil + \lceil \alpha_2 \rceil = 2 \leq \lceil \alpha_1 + \alpha_2 \rceil + 1$, since in this case $\lceil \alpha_1 + \alpha_2 \rceil \in \{1, 2\}$. Together, this proves the assertion.

□

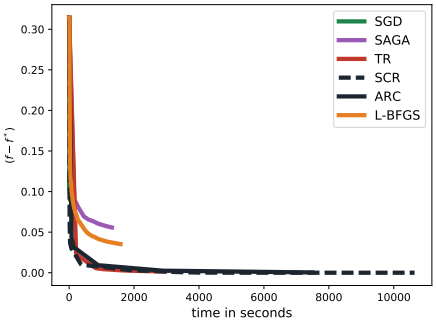
Experimental results



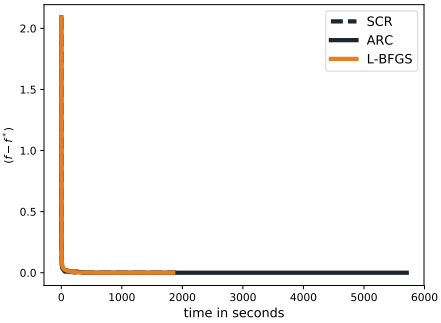
1. log loss cifar



2. log loss mnist



1. loss cifar



2. loss mnist

Figure A.1: (Log) Suboptimality of the empirical risk