Stochastic Second-Order Optimization for Over-parameterized Machine Learning Models

by

Si Yi Meng

B.Sc., University of British Columbia, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

August 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Stochastic Second-Order Optimization for Over-parameterized Machine Learning Models

submitted by **Si Yi Meng** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Mark Schmidt, Computer Science
Supervisor

Michael P. Friedlander, Computer Science
Additional Examiner

Abstract

We consider stochastic second-order methods for minimizing smooth and strongly-convex functions under an interpolation condition, which can be satisfied by over-parameterized machine learning models. Under this condition, we show that the regularized subsampled Newton's method (R-SSN) achieves global linear convergence with an adaptive step-size and a constant batch-size. By growing the batch size for both the subsampled gradient and Hessian, we show that R-SSN can converge at a quadratic rate in a local neighbourhood of the solution. We also show that R-SSN attains local linear convergence for the family of self-concordant functions. Furthermore, we analyze stochastic BFGS algorithms in the interpolation setting and prove their global linear convergence. We empirically evaluate stochastic limited-memory BFGS (L-BFGS) and a "Hessian-free" implementation of R-SSN for binary classification on synthetic, linearly-separable datasets and real datasets under a kernel mapping. Our experimental results demonstrate the fast convergence of these methods, both in terms of the number of iterations and wall-clock time.

Lay Summary

Machine learning applications rely heavily on efficient optimization algorithms to find the best solution minimizing some cost function. By exploiting additional information about the cost function, we show that under certain conditions, one can significantly speed up popular optimization algorithms for models that can completely fit the training data. Faster optimizers allow researchers and practitioners to experiment with more model configurations with less time and resources. The algorithm we analyze is particularly suitable for learning with a large number of training examples, typical in modern-day machine learning.

Preface

This thesis is based on a joint work with Sharan Vaswani, Issam H. Laradji, Mark Schmidt and Simon Lacoste-Julien, appearing at the AISTATS 2020 conference [Meng et al., 2020]. Specifically, I have contributed to proving the theoretical results, setting up and performing the initial experiments on R-SSN with constant and growing batch sizes, as well as comparing related works.

Table of Contents

List of Figures

Glossary

R-SSN regularized subsampled Newton's method

SSN subsampled Newton's method

BFGS Broyden–Fletcher–Goldfarb–Shanno algorithm

L-BFGS limited-memory BFGS

GD gradient descent

SGD stochastic gradient descent

SVRG stochastic variance reduced gradient

LM Levenberg-Marquardt

SVD singular value decomposition

SGC strong growth condition

CG conjugate gradient

RBF radial basis function

GPU graphics processing unit

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Mark Schmidt for inspiring me and guiding me throughout my master's degree. He has given me the opportunity to find the research area that most interests me by encouraging me to explore diverse topics within machine learning research. I would also like to thank my examining committee member Michael P. Friedlander for his time and feedback.

I would like to pay special regards to my close collaborators and mentors: Sharan Vaswani, Issam Laradji, Frederik Kunstner, and Simon Lacoste-Julien for their knowledge and advice. I would like to acknowledge the funding support from NSERC, and I am especially grateful for the detailed feedback from my undergraduate supervisor Kellogg S. Booth on numerous applications.

I have learned so much from my incredible labmates and colleagues, especially Aaron, Alex, Betty, Devon, Jason, Joey, Wilder, and Zixuan. Last but not least, I would like to thank my dear friend Jiaxin, my boyfriend Shou-Chieh, and my parents for helping me and putting up with me during this time. Thank you.

Chapter 1

Introduction

1.1 Optimization in machine learning

Many machine learning tasks can be written as an optimization problem minimizing some objective function $f : \mathbb{R}^d \to \mathbb{R}$. In supervised learning, the objective measures how well the model fits a particular dataset. Training of such models typically translates to finding a solution

$$w^* = \arg\min_w f(w) \quad \text{with} \quad f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w). \tag{1.1}$$

Here, each $f_i(w)$ corresponds to the loss incurred by the model parameterized by w on the example (x_i, y_i) in the training set. We consider the unconstrained setting where the search space for w can be all of $\mathbb{R}^d$.

Assuming differentiability of f, we can solve the above problem using deterministic gradient-based algorithms such as gradient descent (GD) [Cauchy, 1847], which iteratively finds the solution by computing updates of the form

$$w_{k+1} = w_k - \eta_k \nabla f(w_k), \tag{1.2}$$

where $\nabla f(w_k)$ is the gradient of f at w_k. The step size sequence $\{\eta_k\}_{k\geq 0}$ is either defined prior to training or dynamically computed using methods such as backtracking line search. These methods are referred to as *first-order methods*

as they only require first-order derivatives to be computed at every step. First-order methods are popular because the gradient can usually be easily obtained, even for complicated models such as deep neural networks. The computation cost for one iteration is linear in the dimensionality d, which is generally acceptable for typical machine learning models. The convergence behaviour of first-order methods has been well-studied for a wide range of function classes, we refer the reader to Nesterov [2018] for a presentation of some classical analyses. However, when the objective function is ill-conditioned, first-order methods will take a long time to converge. For instance, while GD converges linearly with constant step-size for smooth and strongly-convex functions, the rate of convergence could be rather slow, as illustrated in Fig. 1.1.

The fundamental tradeoff in optimization is often between faster convergence rate and higher iteration cost. Newton's method is well-known for its quadratic convergence rate in a local neighbourhood of the solution by using the following update:

$$w_{k+1} = w_k - \eta_k \left[\nabla^2 f(w_k)\right]^{-1} \nabla f(w_k). \tag{1.3}$$

The Hessian matrix $\nabla^2 f(w_k)$ contains the second-order partial derivatives, which models the local curvature of the objective. Algorithms that incorporate the Hessian are referred to as *second-order methods*. Although quadratic convergence is extremely fast, Newton's update as in Eq. (1.3) is much more expensive to compute, since we need to compute the Hessian and solve a linear system to obtain the update direction, which is cubic in d using standard matrix factorization methods. This cost is practically infeasible for datasets with a large number of features or models with many parameters, both in terms of computation and memory requirement. In this thesis, we mainly restrict our attention to moderate-sized problems where this problem can be partially alleviated using conjugate gradient [Hestenes and Stiefel, 1952] with Hessian-vector products [Pearlmutter, 1994], which allows us to solve for the update in Eq. (1.3) inexactly.

When f has the finite-sum structure as in Eq. (1.1), the gradient and Hessian of f also have a finite-sum structure:

$$\nabla f(w) = \sum_{i=1}^{n} \nabla f_i(w) \quad \text{and} \quad \nabla^2 f(w) = \sum_{i=1}^{n} \nabla^2 f_i(w).$$

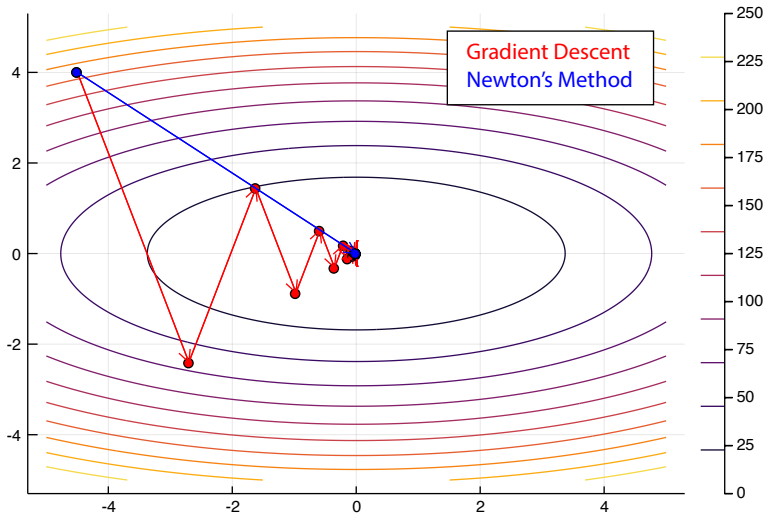

Figure 1.1: Convergence paths taken by gradient descent versus by Newton's method on an ill-conditioned quadratic function. The contours represent the level curves of the objective. While gradient descent takes a zig-zagging route as it converges to the minimum, Newton's method converges in exactly one step.

The training set size n in a typical dataset today can be so large that even parallelized computation of such quantities on a GPU can be prohibitively expensive. In the first-order regime, a natural solution is to replace the full gradient $\nabla f(w)$ with a stochastic approximation $\nabla f_i(w)$, where $i \in \{1, \dots, n\}$ is chosen uniformly at random from the training set. The resulting update

$$w_{k+1} = w_k - \eta_k \nabla f_i(w_k) \tag{1.4}$$

is known as the stochastic gradient descent (SGD) update, introduced by Robbins and Monro [1951]. Alternatively, one can also define a minibatch version as

$$w_{k+1} = w_k - \eta_k \nabla f_{\mathcal{G}}(w_k) \tag{1.5}$$

for some $\mathcal{G} \subset \{1, \dots, n\}$ sampled without replacement. The convergence rate of SGD is usually inferior to its deterministic counterpart, due to the variance in the

stochastic approximation of the gradient $\nabla f_i(w_k)$ or $\nabla f_{\mathcal{G}}(w_k)$. Several variance-reduced versions of SGD have been proposed, including by Defazio et al. [2014], Johnson and Zhang [2013], Schmidt et al. [2017]. However, these methods are still first-order and thus can suffer from ill-conditioning. Popular adaptive methods [Duchi et al., 2011, Kingma and Ba, 2015, Tieleman and Hinton, 2012] alleviate this problem to some extent by using the covariance of stochastic gradients to approximate second order information, although there are no worst-case guarantees on the quality of these approximations. Empirically, these adaptive methods are more robust to the problem's conditioning and result in decent performance across different tasks.

Similarly, in the second-order regime, to reduce the dependence on the number of training examples, subsampled Newton's method (SSN) [Bollapragada et al., 2018a, Erdogdu and Montanari, 2015, Roosta-Khorasani and Mahoney, 2016a,b, Xu et al., 2016] takes a single example or a minibatch to form the subsampled Hessian in each iteration. The corresponding update then becomes

$$w_{k+1} = w_k - \eta_k \left[\nabla^2 f_j(w_k)\right]^{-1} \nabla f_i(w_k), \tag{1.6}$$

for some randomly chosen example i and j that may or may not be the same. The corresponding minibatch version is given by

$$w_{k+1} = w_k - \eta_k \left[\nabla^2 f_{\mathcal{S}}(w_k)\right]^{-1} \nabla f_{\mathcal{G}}(w_k), \tag{1.7}$$

where $\mathcal{G}, \mathcal{S} \subset \{1, \ldots, n\}$ are uniformly chosen minibatches. In addition to the variance coming from the gradient approximation, we now also have variance coming from the Hessian approximation $\nabla^2 f_j(w_k)$ or $\nabla^2 f_{\mathcal{S}}(w_k)$, hampering the fast convergence achieved in the deterministic setting. In this work, we are interested in exploring the class of objective functions for which subsampled Newton's methods can perform competitively albeit using stochastic approximations. This will allow us to justify their use in modern machine learning applications that have a huge number of training examples.

1.2 Over-parameterization and optimization

Recently, there has been a growing interest in optimizing over-parameterized machine learning models. These models are often highly expressive with a hypothesis space so large such that they can interpolate the training data. Interpolation in this case means that there exists a solution w^* for the overall objective f that can simultaneously minimize all component losses f_i. Some works have shown that over-parameterization plays a key role in the observed fast convergence of optimization algorithms. In particular, Du et al. [2019] proved that for a shallow but wide enough ReLU network, over-parameterization and random initialization jointly allows GD to converge at a linear rate to the global minimum. Similarly, Allen-Zhu et al. [2019] show that for deep neural networks with wide layers, even SGD can converge globally at a linear rate with a polynomial dependency on the network width.

Altough over-parameterization seems to be a desirable property that could speed up optimization algorithms, one should nonetheless be skeptical about the generalization properties of such overparamterized models: when we find a solution that achieves zero loss on all training examples, aren't we overfitting? Fortunately, this is not the case for many commonly-used machine learning models. For instance, nonparametric kernel regression without regularization can achieve optimality in both the training loss and the statistical sense [Belkin et al., 2019, Liang and Rakhlin, 2018]. The classical boosting technique can also drive the training error to zero while continuing to decrease the test error [Schapire et al., 1998]. Zhang et al. [2017] empirically demonstrated that for many overparamterized deep neural networks, the gap between the training and test performance is in fact quite small. These results motivate us to investigate optimizers that can train quickly on these highly expressive models but can also generalize well.

Instead of characterizing overparamterization based solely on model size properties such as the number of trainable parameters, another line of work focuses on the interpolation property directly and the implications on the gradients of the objective. If we assume each f_i is differentiable, then interpolation means that each component gradient can also become zero at the solution. In particular, Schmidt and Le Roux [2013] show that with a strong growth condition (SGC) on the size of

the stochastic gradients $\nabla f_i(w)$, SGD with constant step-size achieves the same rate as in the deterministic setting for both convex and strongly-convex functions. Both Ma et al. [2018] and Vaswani et al. [2019a] show that this can be achieved under milder conditions. Additionally, interpolation can allow SGD to match the deterministic rates in the convex [Cevher and Vũ, 2019, Schmidt and Le Roux, 2013, Vaswani et al., 2019a] and non-convex [Bassily et al., 2018, Vaswani et al., 2019a] settings. The interpolation assumption also allows for momentum-type methods to achieve the accelerated rates of convergence for least-squares [Gower et al., 2018] and more generally in convex settings [Liu and Belkin, 2018, Vaswani et al., 2019a]. Although the step size in these settings depends on unknown quantities, it has been recently shown that stochastic line-search methods based on the Armijo condition can be used to automatically set the step size and still achieve fast convergence rates [Vaswani et al., 2019b]. These results are promising in the sense that they demonstrate interpolation can be a favourable condition in speeding up the training of over-parameterized models. However, there has not been studies on how such conditions affect the convergence rate in the second-order regime. One exception is mentioned in Bertsekas [2016], which is the equivalence of the Gauss-Newton method to Newton's method for solving nonlinear least-squares problems when interpolation is satisfied, in which case Gauss-Newton will also enjoy the quadratic convergence in a local neighbourhood of the solution. In this thesis, the main question we aim to answer is how does interpolation play a role in the convergence behaviour of stochastic second-order methods.

1.3 Contributions

We focus on the regularized subsampled Newton's method (R-SSN) that uses the Levenberg-Marquardt (LM) regularization [Levenberg, 1944, Marquardt, 1963] to ensure a well-defined update direction. We first analyze the convergence rate of R-SSN for strongly-convex functions in the interpolation setting. In Chapter 2, we show that R-SSN with an adaptive step-size and a constant batch-size can achieve global linear convergence in expectation. This is in contrast with the work of Bollapragada et al. [2018a] and Bellavia et al. [2018] that analyze subsampled Newton methods without interpolation, and require a geometrically increasing batch-size

to achieve global linear convergence.

If we allow for a growing batch-size in the interpolation setting, R-SSN results in linear-quadratic convergence in a local neighbourhood of the optimal solution. In contrast, in order to obtain superlinear convergence, Bollapragada et al. [2018a] require the batch size for the subsampled gradient to grow at a faster-than-geometric rate. Our results thus show that interpolation allows R-SSN to achieve fast convergence with a more reasonable growth of the batch size.

In Chapter 3, we analyze the convergence of R-SSN for self-concordant functions under the interpolation setting, and show that R-SSN with an adaptive step-size and constant batch-size results in local linear convergence in expectation. Closest to our work is the recent paper by Marteau-Ferey et al. [2019a] that shows approximate Newton methods achieving local linear convergence independent of the condition number; however, they do not consider the interpolation setting and require the approximate Newton directions being "close" to the exact Newton direction with high probability, which is difficult to verify in practice.

In Chapter 4, we view stochastic Quasi-Newton methods as preconditioned SGD and study their behaviour in the interpolation setting. We prove that these algorithms, including the popular L-BFGS [Liu and Nocedal, 1989], the limited-memory version of Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS), can attain global linear convergence with a constant batch-size. Our result is in contrast to previous works that show the global linear convergence of stochastic BFGS algorithms by either using variance-reduction [Kolte et al., 2015, Lucchi et al., 2015, Moritz et al., 2016] or progressive batching strategies [Bollapragada et al., 2018b].

Finally, in Chapter 5, we evaluate R-SSN and stochastic L-BFGS on binary classification problems. We use synthetic linearly-separable datasets and consider real datasets under a kernel mapping. We use automatic differentiation and truncated conjugate gradient [Hestenes and Stiefel, 1952] to develop a "Hessian-free" implementation of R-SSN that allows computing the Newton update without additional memory overhead.[1] When interpolation holds, we observe that both R-SSN and stochastic L-BFGS enjoy faster convergence when compared to popular first-order methods [Duchi et al., 2011, Johnson and Zhang, 2013, Kingma and Ba, 2015].

[1] Our code is available at https://github.com/IssamLaradji/ssn.

Furthermore, a modest batch-growth strategy and stochastic line-search [Vaswani et al., 2019b] scheme ensure that R-SSN is computationally efficient and competitive with stochastic first-order methods and L-BFGS variants in terms of both the number of iterations and wall-clock time required for convergence.

Chapter 2

Subsampled Newton's method

In this chapter, we present our main theoretical results and analyses for R-SSN. We first formally introduce the problem setup and the assumptions we make in Section 2.1, followed by a review of the recent work in the convergence analyses of subsampled Newton-type methods in Section 2.2. We characterize its global linear convergence in Section 2.3 and local quadratic convergence in Section 2.4 under interpolation. We use the notions of Q and R-convergence rates [Nocedal and Wright, 2006, Ortega and Rheinboldt, 1970] reviewed in Appendix A.6. Finally, in Section 2.5, we discuss the future directions that may be of interest.

2.1 Background

We consider the unconstrained minimization of a finite sum as in Eq. (1.1) where the overall objective f and each f_i are twice continuously differentiable. R-SSN takes a step in the subsampled Newton direction at iteration $k \geq 0$:

$$w_{k+1} = w_k - \eta_k \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f_{\mathcal{G}_k}(w_k), \tag{2.1}$$

where η_k is the step size. The index sets $\mathcal{G}_k$ and $\mathcal{S}_k$ are independent samples of indices uniformly chosen from $\{1, \ldots, n\}$ without replacement. We denote the corresponding batch sizes by $b_{g_k} = |\mathcal{G}_k|$ and $b_{s_k} = |\mathcal{S}_k|$. The subsampled gradient

and the regularized subsampled Hessian are defined as

$$\nabla f_{\mathcal{G}_k}(w_k) = \frac{1}{b_{g_k}} \sum_{i \in \mathcal{G}_k} \nabla f_i(w_k), \tag{2.2}$$

$$\mathbf{H}_{\mathcal{S}_k}(w_k) = \frac{1}{b_{s_k}} \sum_{i \in \mathcal{S}_k} \nabla^2 f_i(w_k) + \tau I_d \tag{2.3}$$

where $\tau \geq 0$ is a hyperparameter for the LM regularization [Levenberg, 1944, Marquardt, 1963]. This is also sometimes referred to as the damping parameter in the literature [Martens, 2010] and is related to the inverse of the trust-region radius [Nocedal and Wright, 2006]. Both the subsampled gradient and Hessian (without the regularization) are unbiased, namely

$$\mathbb{E}_{\mathcal{G}_k}[\nabla f_{\mathcal{G}_k}(w_k)] = \nabla f(w_k) \tag{2.4}$$

$$\mathbb{E}_{\mathcal{S}_k}[\mathbf{H}_{\mathcal{S}_k}(w_k)] = \nabla^2 f(w_k) + \tau I_d. \tag{2.5}$$

The independence assumption on $\mathcal{G}_k$ and $\mathcal{S}_k$ is only needed for the analysis and in practice one can simply use the same batch to compute the subsampled gradient and Hessian.

Throughout this thesis, we use $\|\cdot\|$ for the ℓ_2 norm of a vector or the spectral norm of a matrix. For all our convergence results, we make the following standard assumptions:

Assumption 1 (Strong convexity)**.** *The function f in Eq.* (1.1) *is μ-strongly convex such that for all $x, y \in \mathbb{R}^d$ and some $\mu > 0$,*

$$f(y) \geq f(x) + \langle \nabla f(x)\,,\, y - x \rangle + \frac{\mu}{2} \|x - y\|^2 . \tag{2.6}$$

Furthermore, each component f_i is μ_i-strongly convex with $\mu_i \geq 0$.

When $\mu_i = 0$ for some $i \in \{1, \ldots, n\}$, the component function f_i is convex but not strongly convex.

Assumption 2 (Smoothness)**.** *The function f in Eq.* (1.1) *is L-smooth, meaning that the gradient of f is Lipschitz-continuous. Formally, for all $x, y \in \mathbb{R}^d$ and*

some finite L,

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \tag{2.7}$$

which implies the descent inequality [Nesterov, 2018, Lemma 1.2.3]

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|x - y\|^2. \tag{2.8}$$

Furthermore, each component f_i is L_i-smooth with finite L_i.

We denote the condition number of f by $\kappa = L/\mu$. Note that μ_i can be zero as long as μ is strictly positive. This means that the component functions are allowed to only be convex rather than all strongly-convex, in which case the problem would become trivial as all components will be minimized at one unique point, so we would only need to perform optimization on one example. Furthermore, we define

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mu_i \ \ (\leq \mu) \quad \text{and} \quad \bar{L} = \frac{1}{n} \sum_{i=1}^{n} L_i \ \ (\geq L) \tag{2.9}$$

to be the average strong-convexity and smoothness constants of f. These assumptions imply that for any subsample S, the average f_i within S would be L_S-smooth and μ_S-strongly-convex, hence the eigenvalues of the corresponding subsampled Hessian can be upper and lower-bounded. In particular, if we let

$$\tilde{\mu} = \min_S \mu_S \quad \text{and} \quad \tilde{L} = \max_S L_S, \tag{2.10}$$

then for any sample S and point w, the regularized subsampled Hessian $\mathbf{H}_S(w)$ has eigenvalues bounded in the range $[\tilde{\mu} + \tau, \tilde{L} + \tau]$. The LM regularization thus ensures that $\mathbf{H}_S(w)$ will always be positive definite, the subsampled Newton direction exists and is unique.

Since f is strongly-convex, it has a unique minimizer, which we denote by w^*. We focus on over-parameterized machine learning models capable of interpolating the training data, which means all component functions f_i are minimized at w^*. For differentiable functions in the finite sum setting, interpolation formally implies that if $\nabla f(w^*) = 0$, then $\nabla f_i(w^*) = 0$ for all training examples i [Bassily et al., 2018, Ma et al., 2018, Vaswani et al., 2019a,b]. It has been shown that for smooth and

strongly-convex functions, interpolation implies the following growth assumption on the stochastic gradients [Vaswani et al., 2019a, Propositions 1, 2], [Schmidt and Le Roux, 2013].

Assumption 3 (SGC)**.** *A differentiable function f with a finite-sum structure satisfies the strong growth condition (*SGC*) if there exists $\rho \geq 1$ such that for all w,*

$$\mathbb{E}_i \left\|\nabla f_i(w)\right\|^2 \leq \rho \left\|\nabla f(w)\right\|^2.$$

Intuitively, this condition ensures that the stochastic gradients will decrease in expectation as we approach w^*, at which we have $\nabla f(w^*) = 0$. This can be seen as a form of variance reduction inherent to the objective function. As examples, if the training data spans the feature space, then the SGC is satisfied for models interpolating the data when using the squared loss (for regression) or the squared hinge loss (for classification).

2.2 Related work

There has been a rich body of work studying the performance of stochastic second-order methods in machine learning. For SSN-type methods, Roosta-Khorasani and Mahoney [2016a,b] derived probabilistic global and local convergence rates by using matrix concentration inequalities to bound the positive-definiteness of the subsampled Hessian. Xu et al. [2016] extended these works and achieve a faster local linear-quadratic convergence in high probability by using non-uniform sampling to select the batch. Byrd et al. [2011] proposed to use Newton-CG with subsampled Hessian and provide its convergence analysis as well as performance evaluation on a speech recognition application. Pilanci and Wainwright [2017] introduced the Newton Sketch method that uses a random projection of the subsampled Hessian as the Hessian approximation. While Erdogdu and Montanari [2015]'s NewSamp algorithm uses a regularized truncated SVD on the subsampled Hessian to form its inverse, Agarwal et al. [2017] proposed to directly approximate the subsampled Hessian inverse in their algorithm called LiSSA. A unified convergence rate analysis of the above works is provided in Ye et al. [2017], and Li et al. [2020] theoretically justifies the use of these SSN-type methods for high-dimensional data.

Note that in most of these works, although the Hessian is computed (or approximated) using a subsample of the training set, the gradient used in the update is computed exactly. When n is extremely large, these results can be less practically interesting.

In the fully stochastic setting that we consider, Bollapragada et al. [2018a] gave global R-linear rate and local superlinear convergence of SSN in expectation. Their results depend on a batch size growth strategy that may be too aggressive to be practical. Byrd et al. [2012] and Bellavia et al. [2018] obtained similar results for inexact Newton-CG with a heavier emphasis on choosing the Hessian batch size, as well as the forcing sequence to control CG's accuracy at every iteration. For the problem of online linear regression in particular, Patel [2016] introduced the Kalman-based SGD that accumulates an approximation of the inverse Hessian to achieve asymptotic optimality.

Although we only focus on convex objectives in this thesis, in the nonconvex setting, Bergou et al. [2018] showed second-order results for the global convergence of subsampled Newton's method with line search. Milzarek et al. [2019] combined SSN with proximal gradient to solve nonsmooth and noncovex problems. Becker and Le Cun [1988] suggested using a computationally efficient diagonal approximation to the Hessian to improve the performance of second-order methods in neural networks. There is also a line of work that exploit second-order information via the Fisher information matrix as in natural gradient descent [Amari, 1998, Le Roux et al., 2007, Martens and Grosse, 2015, Pascanu and Bengio, 2014] that has been shown to work well in practice for deep learning applications. In contrast to previous works on stochastic second-order methods, we consider R-SSN where both the gradient and the Hessian are subsampled, and show global Q-linear convergence and local Q-quadratic convergence in expectation for SSN for strongly-convex objectives under the interpolation condition.

2.3 Global linear convergence

In this section, we show that for smooth and strongly-convex functions satisfying the interpolation condition, R-SSN with an adaptive step-size and constant batch-sizes for both the subsampled gradient and Hessian converges linearly from an

arbitrary initialization.

Theorem 1 (Global linear convergence)**.** *Under μ-strong convexity, L-smoothness, and ρ-*SGC*, the sequence $\{w_k\}_{k\geq 0}$ generated by* R-SSN *with step size*

$$\eta_k = \frac{\left(\mu_{\mathcal{S}_k} + \tau\right)^2}{L\left(\left(\mu_{\mathcal{S}_k} + \tau\right) + \left(L_{\mathcal{S}_k} + \tau\right) c_g\right)} \tag{2.11}$$

and constant batch sizes $b_{s_k} = b_s$, $b_{g_k} = b_g$ converges to w^* at a Q-linear rate from an arbitrary initialization w_0,

$$\mathbb{E}[f(w_T)] - f(w^*) \leq (1-\alpha)^T \left(f(w_0) - f(w^*)\right), \tag{2.12}$$

where the constants are given by

$$\alpha = \min\left\{\frac{(\bar{\mu}+\tau)^2}{2\kappa c_g(\tilde{L}+\tau)}, \frac{(\bar{\mu}+\tau)}{2\kappa(\tilde{L}+\tau)}\right\} \quad \textit{and} \quad c_g = \frac{(\rho-1)(n-b_g)}{(n-1)\, b_g}. \tag{2.13}$$

The proof is given in Appendix A.2. While the dependence on b_g is explicit, the dependence on b_s is through the constant $\tilde{\mu}$; as b_s tends to n, $\mu_{\mathcal{S}_k}$ tends to μ, allowing R-SSN to use a larger step-size. Since $\mu \geq \bar{\mu}$, the rate characterization constant α will be larger so R-SSN will converge faster. If we set $b_g = b_s = n$ and $\tau = 0$, we obtain the rate

$$\mathbb{E}[f(w_T)] - f(w^*) \leq \left(1 - \frac{1}{2\kappa^2}\right)^T \left[\mathbb{E}[f(w_0)] - f(w^*)\right], \tag{2.14}$$

which matches the deterministic rate [Karimireddy et al., 2018, Theorem 2] up to a factor of 2. The minimum batch-size required to achieve the deterministic rate is

$$b_g \geq \frac{n}{1 + \frac{(\bar{\mu}+\tau)(n-1)}{(\rho-1)}}. \tag{2.15}$$

Similar to SGD [Schmidt and Le Roux, 2013, Vaswani et al., 2019a], the interpolation condition allows R-SSN with a constant batch-size to obtain Q-linear convergence. In the absence of interpolation, SSN has only been shown to achieve an R-linear rate by increasing the batch size geometrically for the subsampled gradient [Bollapragada et al., 2018a]. Next, we analyze the convergence properties of

R-SSN in a local neighbourhood of the solution.

2.4 Local convergence

To analyze the local convergence of R-SSN, we make additional assumptions.

Assumption 4 (Lipschitz continuous Hessian)**.** *The function f in Eq.* (1.1) *has M-Lipschitz continuous Hessian such that for all $x, y \in \mathbb{R}^d$,*

$$\left\|\nabla^2 f(y) - \nabla^2 f(y)\right\| \leq M \left\|x - y\right\|. \tag{2.16}$$

This is a common assumption in analyzing local convergence of second order methods [Bollapragada et al., 2018a, Nesterov, 2018, Roosta-Khorasani and Mahoney, 2016b], and as Ye et al. [2017] have found this is in fact a necessary condition to obtain quadratic convergence.

Assumption 5 (Bounded moments of the iterates)**.** *The sequence $\{w_k\}_{k\geq 0}$ generated by* R-SSN *satisfies*

$$\mathbb{E}\left[\|w_k - w^*\|^2\right] \leq \gamma \left[\mathbb{E}\left\|w_k - w^*\right\|\right]^2 \tag{2.17}$$

for some $0 < \gamma < \infty$, where the expectation is taken over the entire history up until step k.

If the iterates lie within a bounded set, then this assumption holds for some finite γ [Berahas et al., 2020, Bollapragada et al., 2018a, Harikandeh et al., 2015].

Assumption 6 (Bounded variance of the subsampled Hessian)**.** *For all $w \in \mathbb{R}^d$, the subsampled Hessian of the function f in Eq.* (1.1) *has bounded sample variance, namely*

$$\frac{1}{n-1}\sum_{i=1}^{n}\left\|\nabla^2 f_i(w) - \nabla^2 f(w)\right\|^2 \leq \sigma^2 \tag{2.18}$$

for some $\sigma > 0$.

Theorem 2 (Local convergence)**.** *Suppose Assumptions 1 to 3 in Theorem 1 are satisfied. Additionally, under the M-Lipschitz continuous Hessian, γ-bounded mo-*

ments of the iterates, and σ^2-bounded variance of the subsampled Hessian assumptions, the sequence $\{w_k\}_{k\geq 0}$ generated by R-SSN *with unit step-size $\eta_k = 1$ and growing batch-sizes satisfying*

$$b_{g_k} \geq \frac{n}{(\frac{n-1}{\rho-1})\left\|\nabla f(w_k)\right\|^2 + 1}, \quad b_{s_k} \geq \frac{n}{\frac{n}{\sigma^2}\left\|\nabla f(w_k)\right\| + 1} \tag{2.19}$$

converges to w^* at a linear-quadratic rate

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \frac{\gamma(M + 2L + 2L^2)}{2(\tilde{\mu} + \tau)}\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2 + \frac{\tau}{\tilde{\mu} + \tau}\mathbb{E}\left\|w_k - w^*\right\| \tag{2.20}$$

in a local neighbourhood of the solution where $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau)}{\gamma(M+2L+2L^2)}$. Furthermore, when $\tau = 0$ and $\tilde{\mu} > 0$, R-SSN *can achieve local quadratic convergence*

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \gamma\Big(\frac{M + 2L + 2L^2}{2\tilde{\mu}}\Big)\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2.$$

The proof is provided in Appendix A.3. Note that the constant $\frac{\tau}{\tilde{\mu}+\tau}$ on the linear term is less than one and hence contraction is guaranteed, while the constant on the quadratic term is not required to be less than one to guarantee convergence [Nesterov, 2018].

This theorem states that if we progressively increase the batch size for both the subsampled Hessian and the subsampled gradient, then we can obtain a local linear-quadratic convergence rate. For inexact Newton methods to obtain quadratic convergence, it has been shown that the error term needs to decrease as a quadratic function of the gradient [Dembo et al., 1982], which is not provided by SGC, and hence the need for additional techniques such as batch growing. The required geometric growth rate for $\mathcal{G}_k$ is the same as that of SGD to obtain linear convergence without variance-reduction or interpolation [De et al., 2016, Friedlander and Schmidt, 2012]. Note that the proof can be easily modified to obtain a slightly worse linear-quadratic rate when using a subsampled Hessian with a constant batch-size. In addition, following Ye et al. [2017], we can relax the Lipschitz Hessian assumption and obtain a slower superlinear convergence. Unlike the explicit quadratic rate above, in the absence of interpolation, SSN without regulariza-

tion has only been shown to achieve an asymptotic superlinear rate [Bollapragada et al., 2018a]. Moreover, in this case, b_{g_k} needs to increase at a rate that is faster than geometric, considerably restricting its practical applicability.

The following corollary (proved in Appendix A.3.1) states that if we decay the LM-regularization sequence τ_k proportional to the gradient norm, R-SSN can achieve quadratic convergence for strongly-convex functions. In fact, this decay rate is inversely proportional to the growth of the batch size for the subsampled Hessian, indicating that larger batch-sizes require smaller regularization. This is expected because our overall objective f is strongly-convex, as we increase the batch size the subsampled Hessian is closer to being positive-definite, and therefore less regularization is required. This relationship between the regularization and sample size is also consistent with the findings of Ye et al. [2017].

Corollary 1 (Local quadratic convergence)**.** *Under the same assumptions as Theorem 2, if we decrease the regularization term according to* $\tau_k \leq \|\nabla f(w_k)\|$*,* R-SSN *can achieve local quadratic convergence with* $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau_k)}{\gamma(M+4L+2L^2)}$*:*

$$\mathbb{E}\,\|w_{k+1} - w^*\| \leq \frac{\gamma(M + 4L + 2L^2)}{2(\tilde{\mu}_k + \tau_k)}\,\left(\mathbb{E}\,\|w_k - w^*\|\right)^2,$$

where $\tilde{\mu}_k$ *is the minimum eigenvalue of* $\nabla^2 f_{\mathcal{S}_k}(w_k)$ *over all batches of size* $|\mathcal{S}_k|$*.*

Since $\tilde{\mu} \leq \tilde{\mu}_k$, this gives a tighter guarantee on the decrease in suboptimality at every iteration as the batch size grows and τ_k decreases. On the other hand, if we make a stronger growth assumption on the stochastic gradients such that $\mathbb{E}_i\,\|\nabla f_i(w)\|^2 \leq \rho\,\|\nabla f(w)\|^4$, then R-SSN can achieve local quadratic convergence using only a *constant* batch-size for the subsampled gradient and the same growth rate for the subsampled Hessian. We state this result as Corollary 2 in Appendix A.3.2, but we acknowledge that this assumption might be too restrictive to be useful in practice.

2.5 Future work

Convexity of the component functions are crucial to R-SSN as they guarantee the eigenvalues of the subsampled Hessians are at least 0, and thus we only need to

add an LM-regularization to ensure the overall update is well-defined. As many modern machine learning models lead to nonconvex objectives such as deep neural networks, ideally we would like to devise algorithms that can gracefully handle negative curvature with provable fast convergence. In the nonconvex setting, it may be interesting to analyze stochastic Gauss-Newton methods under interpolation where the Gauss-Newton matrix is constructed using subsampled gradients but still remains positive semidefinte.

Although our local convergence analysis yields faster rates with a less aggressive batch growth compared to previous works, increasing the batch size at every iteration is still not an ideal strategy in practice. Not only does the growth rate need to be tuned, machine learning practitioners would need to modify the training loop in their implementation to incorporate this change. Furthermore, increasing the batch size can also lead to more expensive iterations and larger memory requirements. Exploring other techniques to essentially reduce the variance in the update direction without incurring significant computational overhead and implementation burden can potentially lead to wider practical acceptance of R-SSN.

Another interesting direction is instead of using the subsampled Hessians as the Hessian approximation, one may attempt to analyze the convergence rates using subsampled diagonal or block-diagonal approximations to the Hessian. These alternatives are favorable because solving for the update direction would be much cheaper, and they can also significantly reduce the storage requirement due to the sparsity compared to using the full, subsampled Hessian. Although the analysis would naturally fall into the preconditioned SGD analysis, which we will discuss in Chapter 4, one may be able to leverage the second-order information in these approximations to obtain faster rates.

Chapter 3

Self-concordance

It is easy to show that if we transform the iterates using a nonsingular linear operator, the corresponding Newton update is the same as that of in the original space under the same transformation. This property of Newton's method is known as affine invariance. However, it is not reflected in classical analysis in which an affine transformation on the iterates will cause the condition number to change, thereby changing the rate of convergence. In addition, the strong convexity, first and second-order smoothness constants are often unknown in practice. Self-concordance, as we will explain next, is a property of a function that can be used to yield affine-invariant rate for Newton's method. In this chapter, we present our analysis of R-SSN for functions within this class, although our rate still bears problem-dependent constants and it is unclear how to remove this dependence in the stochastic setting.

3.1 Background

We first introduce the definition of self-concordance in the univariate case.

Definition 1 ([Boyd and Vandenberghe, 2004])**.** *A convex function $f : \mathbb{R} \to \mathbb{R}$ is self-concordant if for all $x \in \mathbb{R}$*

$$|f'''(x)| \leq 2f''(x)^{3/2}. \tag{3.1}$$

The constant 2 in the above definition is arbitrary, and we can choose any positive constant such that an appropriately rescaled f will satisfy Definition 1. In-

tuitively, instead of using problem-specific constants to prescribe a bound on the curvature's rate of change, self-concordance allows one to do so using the curvature itself to guarantee the Hessian doesn't chang too rapidly. It is easy to show that self-concordance is preserved under an affine transformation of the input, which makes it a desirable alternative to the strong-convexity and smoothness assumptions. Examples of self-concordant functions include linear and quadratic functions, since the third derivatives are zero. Negative logarithm and its sum with negative entropy are also classic examples commonly used in interior-point methods. In the multivariate setting, one defines self-concordance as the following:

Definition 2 ([Boyd and Vandenberghe, 2004])**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is self-concordant if it is self-concordant along every line, i.e., if the function $g(t) = g(x + tv)$ is self-concordant by Definition 1 for all t and $x, v \in \mathbb{R}^d$.*

An equivalent definition using tensor derivatives is given below, which is more similar to the univariate case as in Definition 1.

Definition 3 ([Nesterov and Nemirovskii, 1994])**.** *A function $f : \mathbb{R}^d \to \mathbb{R}$ is self-concordant if for all $x, h \in \mathbb{R}^d$,*

$$|D^3 f(x)[h, h, h]| \leq 2(D^2(x)[h, h])^{3/2}, \tag{3.2}$$

where $D^k(x)[h_1, h_2, \ldots, h_k]$ denotes the kth differential of f at x along the directions $h_1, h_2, \ldots, h_k$.

Similar to convexity, there exists operations under which self-concordance is preserved. Multiplicative scaling of a self-concordant function by a factor greater than 1 preserves self-concordance, the sum of two self-concordant functions is self-concordant, and composition with affine functions or logarithm is also self-concordant [Boyd and Vandenberghe, 2004]. The analysis for Newton's method for self-concordant functions relies heavily on a quantity called the Newton decrement:

$$\lambda(w) := \left\langle \nabla f(w)\,,\, \left[\nabla^2 f(w)\right]^{-1} \nabla f(x) \right\rangle^{1/2} = \|\nabla f(w)\|_{[\nabla^2 f(w)]^{-1}}\,. \tag{3.3}$$

The Newton decrement is essentially measuring the (quadratic) norm of the gradient with respect to the local curvature defined by the Hessian, and therefore it

is affine-invariant. Similar to the Euclidean norm of the gradient, the Newton decrement can be used as a stopping criterion in Newton's method. Alternative to measuring progress using the suboptimality gap between the current function value or iterate to that of the optimum, we can instead use the Newton decrement and obtain similar bounds without problem-specific constants. A reference to such analysis for Newton's method can be found in Boyd and Vandenberghe [2004].

3.2 Related work

Self-concordance was first introduced by Nesterov and Nemirovskii [1994] for barrier functions in interior-point methods. The least-squares loss naturally satisfies the self-concordance property because its third derivatives are 0. Other common machine learning losses may require slight modifications on the definition. Bach [2010] removed the power on the second derivative so that the condition can be satisfied by logistic regression, and shows that under this modified condition, Newton's method achieves convergence rate comparable to that obtained under standard self-concordance. However, this result does not preserve affine-invariance. Marteau-Ferey et al. [2019b] further generalized the definition of Bach [2010], which allows the Huber loss and generalized linear models to be included in this functional class. For ℓ_2-regularized losses, Zhang and Lin [2015] showed that if the third derivative of the overall loss can be bounded by the second to some power between 0 and 1, then it is self-concordant by Definition 1. They show that regularized logistic regression and smoothed hinge loss belong to this class. Essentially all these extended definitions boils down to different ways of controlling the third derivative by the second.

Recently, there has been growing interests in analyzing variants of Newton's method for self-concordant functions. Pilanci and Wainwright [2017] provided global convergence results for Newton Sketch for functions within this class. Zhang and Lin [2015] proposed a distributed inexact Newton's method in which the master node solves for the Newton update using inexact conjugate gradient (CG). Our analysis is similar to theirs but we instead adapt the step size and analysis to the stochastic setting. Eisen et al. [2018], Mokhtari et al. [2016] introduced the Ada Newton method that computes a geometrically increasing batch-size based

on a targeted statistical accuracy, and show local quadratic convergence for self-concordant functions with high probability. However, none of these works analyze stochastic Newton methods for self-concordant functions. Marteau-Ferey et al. [2019a] showed that for functions in the generalized self-concordance class [Bach, 2010], approximate Newton methods can achieve local linear convergence independent of the condition number; however, they require the approximate Newton directions being "close" to the exact Newton direction with high probability.

3.3 Stochastic formulation

To characterize the convergence rate of R-SSN under self-concordance, we define the regularized Newton decrement at w by

$$\lambda(w) := \langle \nabla f(w)\,,\, \left[\nabla^2 f(w) + \tau I_d\right]^{-1} \nabla f(w) \rangle^{1/2}. \tag{3.4}$$

When $\tau = 0$, this corresponds to the standard definition of the Newton decrement in Eq. (3.3). If we denote the standard Newton decrement as $\lambda^0(w)$, then the regularized version can be bounded as $\lambda(w) \leq \lambda^0(w)$. We also introduce the regularized stochastic Newton decrement as

$$\tilde{\lambda}_{i,j}(w) := \langle \nabla f_i(w)\,,\, [\mathbf{H}_j(w)]^{-1} \nabla f_i(w) \rangle^{1/2} \tag{3.5}$$

for independent samples i and j, where $\mathbf{H}_j(w)$ is the regularized subsampled Hessian defined in Eq. (2.3). Again, the fact that we use independent samples for the subsampled gradient and Hessian here is only a technicality of the analysis.

3.4 Convergence analysis

In the following proposition, we show that a similar growth condition for the regularized stochastic Newton decrement can be derived from the SGC (Assumption 3).

Proposition 1. *Suppose function f satisfies the* SGC *(Assumption 3) with parameter ρ. For any positive definite matrices A, B with extreme eigenvalues $\lambda_{\min}(B)$*

and $\lambda_{\max}(A)$, the following inequality holds for all $w \in \mathbb{R}^d$:

$$\mathbb{E}\,\|\nabla f_{\mathcal{G}}(w)\|_A^2 \leq \frac{\rho\lambda_{\max}(A)}{\lambda_{\min}(B)}\,\|\nabla f(w)\|_B^2\,. \tag{3.6}$$

Proof. From the LHS, we have

$$\mathbb{E}\,\|\nabla f_{\mathcal{G}}(w)\|_A^2 \leq \lambda_{\max}(A)\mathbb{E}\,\|\nabla f_{\mathcal{G}}(w)\|^2 \leq \rho\lambda_{\max}(A)\,\|\nabla f(w)\|^2\,.$$

From the RHS, we have

$$\|\nabla f(w)\|_B^2 \geq \lambda_{\min}(B)\,\|\nabla f(w)\|^2\,.$$

Combining the two inequalities gives us the desired result. □

Applying this proposition to the regularized stochastic Newton decrement in Eq. (3.5) gives us the following assumption.

Assumption 7 (Newton decrement SGC)**.** *For all independent w and j, there exists $\rho_{nd} \geq 1$ such that*

$$\mathbb{E}_i\left[\tilde{\lambda}_{i,j}^2(w)\right] \leq \rho_{nd}\,\lambda^2(w). \tag{3.7}$$

For ease of notation, we use $\tilde{\lambda}_k$ to denote the regularized stochastic Newton decrement computed using samples $\mathcal{G}_k$ and $\mathcal{S}_k$. We now consider an update step similar to Eq. (2.1) but with the step size adjusted to $\tilde{\lambda}_k$,

$$w_{k+1} = w_k - \frac{c\,\eta}{1+\eta\,\tilde{\lambda}_k}\,[\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\,\nabla f_{\mathcal{G}_k}(w_k) \tag{3.8}$$

where $c, \eta \in (0, 1]$. For the next theorem, we assume the iterates lie in a bounded set [Harikandeh et al., 2015] such that $\|w_k - w^*\| \leq D$ for all k.

Theorem 3 (Two-phased analysis)**.** *Suppose f is self-concordant and satisfies L-smoothness, and that the subsampled Hessians have bounded eigenvalues in the range $[\tilde{\mu}+\tau, \tilde{L}+\tau]$ with $\tilde{\mu} \geq 0$. Suppose Newton decrement* SGC *holds with parameter $\rho_{nd} = \frac{\rho L}{\tilde{\mu}+\tau}$. Then if the sequence $\{w_k\}_{k\in[0,m]}$ generated by* R-SSN *in*

Eq. (3.8) *stay in the bounded set with radius* D *with*

$$\eta \in \left(0, \frac{c}{\rho_{nd}(1 + \tilde{L}D/(\tilde{\mu} + \tau))}\right] \quad \textit{where} \quad c = \sqrt{\frac{\tilde{\mu} + \tau}{L}}, \tag{3.9}$$

and constant batch sizes converges to w^* *from an arbitrary initialization* w_0 *at a rate characterized by*

$$\mathbb{E}\left[f(w_{k+1})\right] \leq f(w_k) - \eta\, \delta\, \omega\left(\lambda_k\right).$$

Here $\delta \in (0, 1]$ *and the univariate function* ω *is defined as* $\omega\left(t\right) = t - \ln(1 + t)$. *Furthermore, in the local neighbourhood where* $\lambda_m \leq 1/6$, *the sequence* $\{w_k\}_{k \geq m}$ *converges to* w^* *at a Q-linear rate, given by*

$$\mathbb{E}\left[f(w_T)\right] - f(w^*) \leq \left(1 - \frac{\eta\delta}{1.26}\right)^{T-m} (\mathbb{E}\left[f(w_m)\right] - f(w^*)). \tag{3.10}$$

The proof is given in Appendix A.4. The above result gives insight into the convergence properties of R-SSN for loss functions that are self-concordant but not necessarily strongly-convex. Note that the analysis of the deterministic Newton's method for self-concordant functions yields a problem-independent local quadratic convergence rate, meaning it does not rely on the condition number of f. However, the rate we obtain above is still problem-dependent, as both the step size η and the rates themselves depend on $\tilde{\mu}$, $\tilde{L}$, and L. It is non-trivial to improve the above analysis and obtain a similar affine-invariant result. Similar to previous work, the algorithm parameters for the inexact Newton method in Zhang and Lin [2015] also depend on the condition number.

3.5 Future work

An obvious future direction is to search for analyses that yield problem-independent rates and hyperparameters for R-SSN under self-concordance. One may need to modify the way the effective step size adapts to the regularized stochastic Newton decrement, as in Eq. (3.8). Self-concordance analysis for Newton's method in the deterministic case is mainly to address the theoretical and practical discrepancy, and therefore we did not include experiments specifically for R-SSN on self-

concordant functions. However, it would be interesting to see how the adaptive effective step-size in Eq. (3.8) perform in practice, even on objectives that are not necessarily self-concordant. Since the R-SSN update already requires the Hessian-vector product $[\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f_{\mathcal{G}_k}(w_k)$, computing $\tilde{\lambda}_k$ does not incur substantial computation overhead.

Chapter 4

Quasi-Newton methods as preconditioned SGD

4.1 Background

The regularized subsampled Newton's method (R-SSN) algorithm considered in previous chapters can be viewed as preconditioned SGD, where the preconditioner of the stochastic gradient direction is the inverse of the (regularized) subsampled Hessian matrix. In fact, many popular stochastic optimization algorithms used in machine learning can be expressed as some variant of preconditioned SGD, which takes the following general form:

$$w_{k+1} = w_k - \eta_k \mathbf{B}_k \nabla f_{\mathcal{G}_k}(w_k). \tag{4.1}$$

Here, $\{\eta_k\}$ is again the step size sequence and $\nabla f_{\mathcal{G}_k}(w_k)$ is the subsampled gradient. The preconditioner $\mathbf{B}_k$ is a $d \times d$ symmetric, positive-definite matrix usually designed to project the stochastic gradient step such that the resulting update direction adapts to the local curvature of the problem. For SGD, the preconditioner $\mathbf{B}_k$ is simply the identity matrix. If we sum the outer product of the stochastic gradients over all iterations and take its inverse square root, we obtain the AdaGrad preconditioner [Duchi et al., 2011]. In this chapter, we are particularly interested in quasi-Newton methods which use low rank approximations to the Hessian or its

inverse as the preconditioner. The most popular quasi-Newton methods are BFGS [Nocedal and Wright, 2006] and L-BFGS, the limited-memory version [Liu and Nocedal, 1989]. If we compute the full gradient, i.e. $\mathcal{G}_k = \{1, \ldots, n\}$, then the original BFGS update for the preconditioners is

$$\mathbf{B}_{k+1} = (I - \rho_k s_k y_k^T)\mathbf{B}_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \tag{4.2}$$

where $s_k = w_{k+1} - w_k$, $y_k = \nabla f_{\mathcal{G}_{k+1}}(w_{k+1}) - \nabla f_{\mathcal{G}_k}(w_k)$, and $\rho_k = (y_k^T s_k)^{-1}$. The step size used in BFGS is chosen via line search to satisfy the Wolfe conditions, which will guarantee positive-definiteness of $\mathbf{B}_{k+1}$. There is no general rule for choosing the initial preconditioner $\mathbf{B}_0$ in BFGS methods, one option is to just take a positive multiple of the identity matrix. To scale to problems of larger dimensions, instead of storing the full $d \times d$ matrix $\mathbf{B}_k$, the L-BFGS method stores a few past $\{s_k, y_k\}$ pairs such that a modified version of $\mathbf{B}_k \nabla f(w_k)$ can be computed efficiently [Nocedal and Wright, 2006].

In the deterministic case, the BFGS method and other quasi-Newton methods such as the Davidon–Fletcher–Powell (DFP) method enjoy superlinear convergence rates as characterized by Dennis and Moré [1974]. These methods are attractive not only for their fast convergence, but also for their cheaper iteration costs as they only require computing the gradient instead of the Hessian as in Newton's method. In this chapter, we provide analyses for these type of methods in the stochastic setting where the full gradient $\nabla f(w_k)$ is replaced by the subsampled estimates $\nabla f_{\mathcal{G}_k}(w_k)$, specifically under the interpolation condition. Although our experimental results in Chapter 5 focus on the performance of stochastic L-BFGS methods, our theoretical results can apply to any positive-definite preconditioner.

4.2 Related work

As our focus in this chapter is on preconditioned SGD in which the preconditioner specifically tries to approximate the Hessian, we mainly discuss related works on stochastic Quasi-Newton methods. In the determinstic case, Quasi-Newton methods use successive iterates and gradients to construct a positive-definite matrix as the Hessian approximation. Berahas et al. [2019] proposed to sample around the current iterate to construct better approximation to the Hessian inverse, although

their algorithm is still deterministic in terms of the gradient computation. To reduce the dependence on the size of the training set, Schraudolph et al. [2007] extended deterministic quasi-Newton methods to the online convex optimization setting in which the gradients used in their oBFGS and oLBFGS algorithms are stochastic approximations to the full gradient. They show that these methods perform well on non-interpolating quadratic problems and conditional random fields against well-tuned SGD and natural gradient descent. Their empirical results were later backed by convergence results from a unified theoretical framework for online variable metric methods [Sunehag et al., 2009].

One limitation in these subsampled quasi-Newton methods is that the preconditioner update depends on stochastic gradients computed at successive iterations, and more importantly, using different batches. Unless the batch size is large enough to guarantee sufficient overlap, this update can become extremely unstable. To alleviate this issue, and Byrd et al. [2012], Friedlander and Schmidt [2012] and Bollapragada et al. [2018b] proposed to progressively increase the batch size during the course of training, where the batch growth strategy is determined by different tests based on the stochastic gradient variance. These algorithms converge linearly for smooth and strongly-convex objectives and sublinearly for convex objectives. On the other hand, Liu et al. [2018] used approximate second-order information to provide stabilization and showed convergence to a neighbourhood of the solution at a linear rate for both convex and nonconvex objectives.

Combined with variance-reduction techniques from popular first-order methods, stochastic BFGS-type methods can achieve global linear convergence [Kolte et al., 2015, Lucchi et al., 2015, Moritz et al., 2016]. Gower et al. [2016] developed stochastic block BFGS updates that combines sketching strategies and variance reduction to achieve linear convergence. Zhao et al. [2018] proposed a coordinate transformation framework for analyzing the algorithms by Moritz et al. [2016] and Gower et al. [2016], and suggested practical tricks such as adopting non-uniform sampling when computing the stochastic gradients. Furthermore, Chang et al. [2019] incorporated momentum into variance-reduced L-BFGS and achieves an accelerated global linear convergence.

In the superlinear regime, Rodomanov and Kropotov [2016]'s Newton-type Incremental Method (NIM) maintains a quadratic model of the objective using sub-

sampled Hessian information. Mokhtari et al. [2018]'s Incremental Quasi-Newton (IQN) method also maintains aggregated information of the quadratic model, but it does so without requiring Hessian computation nor an inverse. Although both methods can achieve superlinear convergence, they require additional additional memory in exchange for variance reduction. Instead of the commonly-used line-search strategy, adaptive step-size strategies have been proposed to achieve R-superlinear convergence with a growing batch-size [Gao and Goldfarb, 2019, Zhou et al., 2017]. On a slightly different perspective, Kelley [2002]'s implicit filtering method can achieve superlinear convergence using a deterministic and finite-difference-based BFGS for minimizing objectives whose noise decreases as we approach the global minimum, similar to the interpolation setting. This work fills in the gap between stochastic quasi-Newton methods and the interpolation condition for over-parameterized models.

4.3 Convergence analysis

Consider the stochastic BFGS update,

$$w_{k+1} = w_k - \eta_k \, \mathbf{B}_k \, \nabla f_{\mathcal{G}_k}(w_k) \tag{4.3}$$

where $\mathbf{B}_k$ is a positive definite matrix constructed to approximate the inverse Hessian $\left[\nabla^2 f(w_k)\right]^{-1}$. For convex objectives, we can guarantee positive-definiteness by including a small LM-regularization value to the stochastic BFGS preconditioner [Schraudolph et al., 2007]. Therefore, as in previous works [Bollapragada et al., 2018b, Lucchi et al., 2015, Moritz et al., 2016, Sunehag et al., 2009], we assume that $\mathbf{B}_k$ has eigenvalues such that $\lambda_1 I \preceq \mathbf{B}_k \preceq \lambda_d I$ for all k, where $\lambda_1 > 0$. We now show that under the strong growth condition (SGC), stochastic BFGS achieves global linear convergence with a constant step-size.

Theorem 4 (Global linear convergence of stochastic BFGS)**.** *Let μ-strong convexity, L-smoothness, and ρ-SGC be satisfied, and suppose the eigenvalues of $\mathbf{B}_k$ are bounded in $[\lambda_1, \lambda_d]$. Then the sequence $\{w_k\}_{k\geq 0}$ generated by stochastic BFGS with constant step-size $\eta_k = \eta = \frac{\lambda_1}{c_g L \lambda_d^2}$ and constant batch size $b_{g_k} = b_g$ converges*

to w^* at a linear rate from an arbitrary initialization w_0,

$$\mathbb{E}[f(w_T)] - f(w^*) \leq \left(1 - \frac{\mu\lambda_1^2}{c_g\, L\lambda_d^2}\right)^T (f(w_0) - f(w^*))$$

where $c_g = \frac{(n-b_g)\,(\rho-1)}{(n-1)\,b_g} + 1$.

The proof is given in Appendix A.5. This rate matches the global linear rate for R-SSN in Theorem 1 up to constants without having to compute second order information. The strong-convexity assumption can be relaxed to the Polyak-Łojasiewicz inequality [Karimi et al., 2016, Polyak, 1963] while giving a similar rate. In the absence of interpolation, global linear convergence can only be obtained by using either variance-reduction techniques [Lucchi et al., 2015, Moritz et al., 2016] or progressive batching [Bollapragada et al., 2018b]. Similar to these works, our analysis for the stochastic BFGS method applies to preconditioned SGD where $\mathbf{B}_k$ can be any positive-definite preconditioner.

4.4 Future work

As noted previously, our theoretical result in this chapter holds for all positive-definite preconditioners with bounded eigenvalues. The fact that our analysis does not take into account the specific structure of quasi-Newton preconditioners might be the reason why we were unable to obtain a rate superior to that of SGD in the interpolation setting. Recently, Kovalev et al. [2020] showed local linear and superlinear convergence for randomized BFGS for self-concordant functions, as well as a local linear rate for smooth and strongly-convex functions. It may be possible to leverage their proof techniques and seek a globally fast convergence rate specific to quasi-Newton methods under the interpolation condition. Extending these analyses to the convex and non-convex settings would be another interesting direction.

Chapter 5

Experiments

In this chapter, we verify the fast convergence of stochastic second-order methods on convex problems where interpolation is satisfied. Our experiments are performed on a binary classification task using both synthetic and real datasets. We evaluate two variants of R-SSN: `R-SSN-const` that uses a constant batch-size and `R-SSN-grow` where we grow the batch size geometrically (by a constant multiplicative factor of 1.01 in every iteration [Friedlander and Schmidt, 2012]).Although our theoretical analysis of R-SSN requires independent batches for the subsampled gradient and Hessian, we use the same batch for both variants of R-SSN to reduce the computation costs and observe that this does not adversely affect the empirical performance. We use truncated CG [Hestenes and Stiefel, 1952] to solve for the (subsampled) Newton direction in every iteration. For each experiment, we choose the LM regularization τ via a grid search. For `R-SSN-grow`, following Corollary 1, starting from the LM regularization selected by grid search, we progressively decrease τ in the same way as we increase the batch size. We evaluate stochastic L-BFGS (`sLBFGS`) with a "memory" of 10. For `sLBFGS`, we use the same minibatch to compute the difference in the (subsampled) gradients to be used in the inverse Hessian approximation (this corresponds to the "full" overlap setting in Bollapragada et al. [2018b]), to which we add a small regularization to ensure positive-definiteness. For both R-SSN and `sLBFGS`, we use the stochastic line-search from Vaswani et al. [2019b] to set the step size.

We compare the proposed algorithms against common first-order methods:

SGD, SVRG [Johnson and Zhang, 2013], Adam [Kingma and Ba, 2015] and AdaGrad [Duchi et al., 2011]. For all experiments, we use an (initial) batch size of $b = 100$ and run each algorithm for 200 epochs. Here, an epoch is defined as one full pass over the dataset and does not include additional function evaluations from the line-search or CG. Subsequently, in Figs. 5.2 to 5.4, we plot the mean wall-clock time per epoch that takes these additional computations into account. For SGD, we compare against its generic version and with Polyak acceleration [Polyak, 1964], where in both cases the step size is chosen via stochastic line-search [Vaswani et al., 2019b] with the same hyperparameters, and the acceleration hyperparameter is chosen via grid search. For SVRG, the step size is chosen via 3-fold cross validation on the training set, and we set the number of inner iterations per outer full-gradient evaluation to n/b. We use the default hyperparameters for Adam and AdaGrad for their adaptive properties. All results are averaged across 5 runs.

5.1 Synthetic and linearly-separable datasets

We first evaluate the algorithms on binary classification using synthetic, linearly-separable datasets with varying margins. Linear separability ensures the interpolation condition will hold with a linear model. For each margin, we generate a dataset with 10k examples with $d = 20$ features and binary labels. For these datasets, we also compare against unregularized Newton and L-BFGS, both using the full-batch (hence deterministic). For L-BFGS, we use the PyTorch [Paszke et al., 2019] implementation with line search and an initial step size of 0.9.

In Fig. 5.1, we show the training loss for the logistic loss (row 1) and the squared hinge loss (row 2). We observe that by incorporating second-order information, R-SSN can converge much faster than first order methods. In addition, global linear convergence can be obtained using only constant batch-sizes, verifying Theorem 1. Furthermore, by growing the batch size, R-SSN performs similar to the deterministic Newton method, verifying the local quadratic convergence of Theorem 2. Although for stochastic L-BFGS, our theory only guarantees linear convergence globally (instead of superlinear) under interpolation, these methods can be much faster than first order methods empirically. Finally, as we increase the margin (from left to right of Fig. 5.1), the theoretical rate under interpolation

improves since ρ decreases, resulting in faster and more stable convergence for the proposed algorithms.

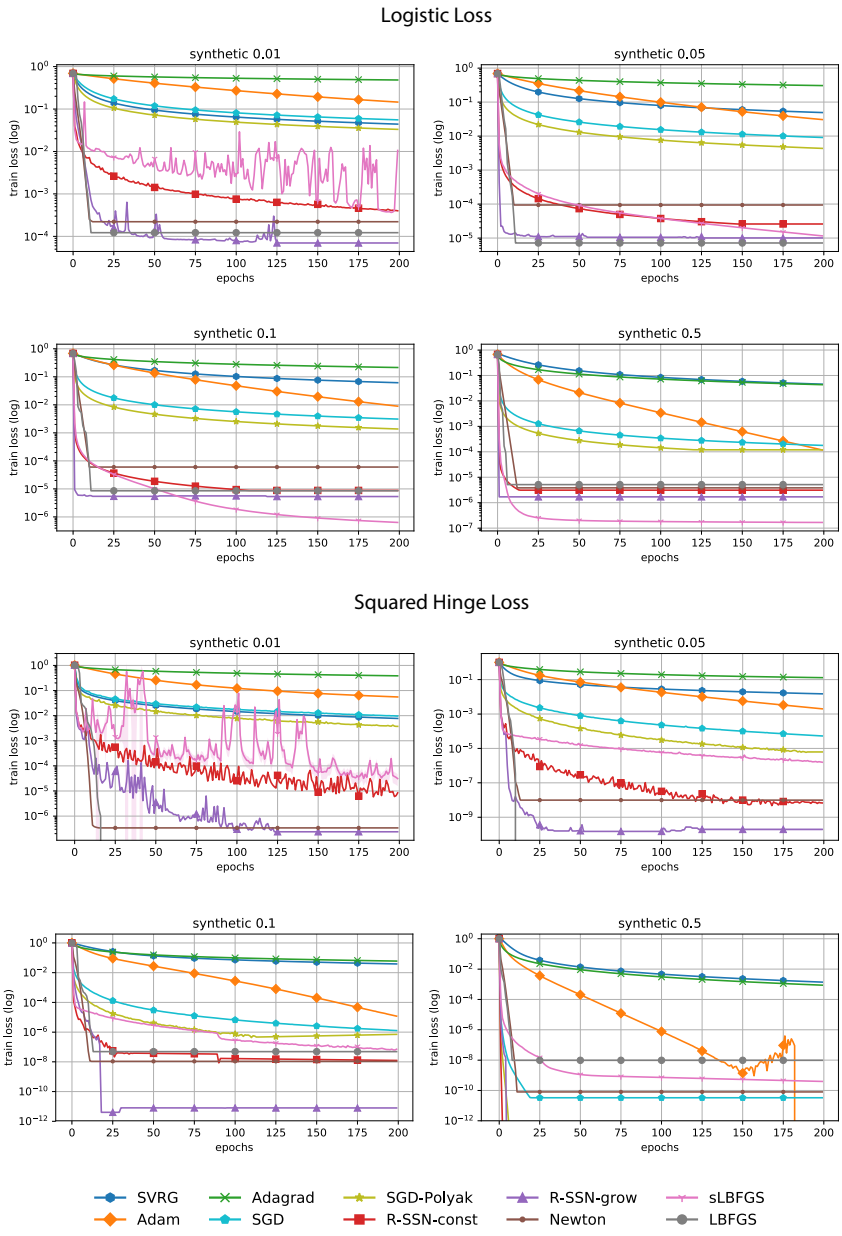

Figure 5.1: Comparison of R-SSN variants and stochastic L-BFGS against first order methods on synthetic data where interpolation is satisfied, and both R-SSN outperform first order methods. For each loss, the results are on datasets with linearly-separable margins in $[0.01,\ 0.05,\ 0.1,\ 0.5]$.

5.2 Real datasets

We also consider real datasets `mushrooms`, `rcv1`, and `ijcnn` from the LIBSVM repository [Chang and Lin, 2011], and use an $80:20$ split for the training and test set, respectively. We fit a linear model under the radial basis function (RBF) kernel. The kernel mapping results in the effective dimension being equal to the number of points in the dataset. This results in problem dimensions of 6.5k, 20k and 28k for `mushrooms`, `rcv1`, and `ijcnn` respectively. The RBF-kernel bandwidths are chosen via grid search using 10-fold cross validation on the training split, following Vaswani et al. [2019b]. Note that the `mushrooms` dataset is linearly separable under the chosen kernel mapping, and thus satisfies the interpolation condition. For these datasets, we limit the maximum batch size to 8192 for `SSN-grow` to alleviate the computation and memory overhead. We show the training loss, test accuracy, as well as the mean wall-clock time per epoch. For this set of experiments, the stochastic line-search procedure [Vaswani et al., 2019b] used in conjunction with `sLBFGS` can lead to a large number of backtracking iterations, resulting in a high wall-clock time per epoch. To overcome this issue, we use a constant step-size variant of `sLBFGS` and perform a grid search over $[10^{-4}, 1]$ to select the best step-size for each experiment.

In the first rows of Figs. 5.2 to 5.4, we observe that when interpolation is satisfied (`mushrooms`), the R-SSN variants and `sLBFGS` outperform all other methods in terms of training loss convergence. When interpolation is not satisfied (`ijcnn` and `rcv1`), the stochastic second-order methods are still competitive with the best performing method. In the second row, we observe that despite the fast convergence of these methods, generalization performance does not deteriorate regardless of whether interpolation is satisfied. Furthermore, since all our experiments are run on the GPU, we take advantage of parallelization and amortize the runtime of the proposed methods. This is reflected in the third row, where we plot the mean per-epoch wall-clock time for all methods. Also note that `sLBFGS` is competitive with R-SSN in terms of the number of iterations, but has a higher per-epoch cost on average.

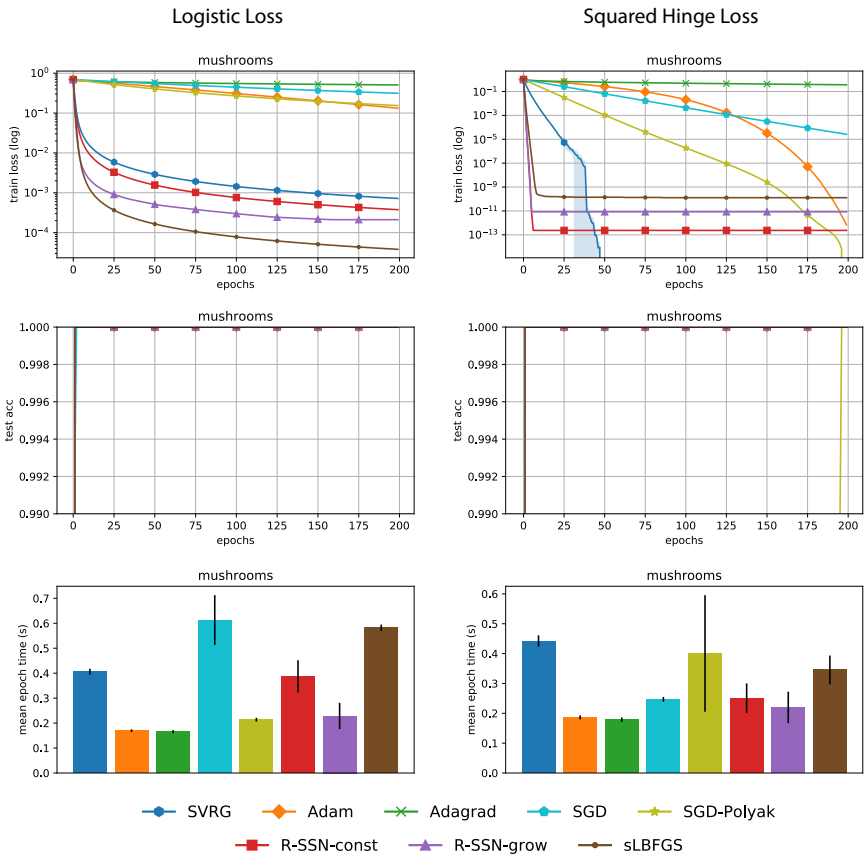

Figure 5.2: Comparison of R-SSN variants and stochastic L-BFGS against first-order methods on the `mushrooms` dataset, which is linearly-separable under the RBF kernel. R-SSN variants and `sLBFGS` perform the best in this setting.

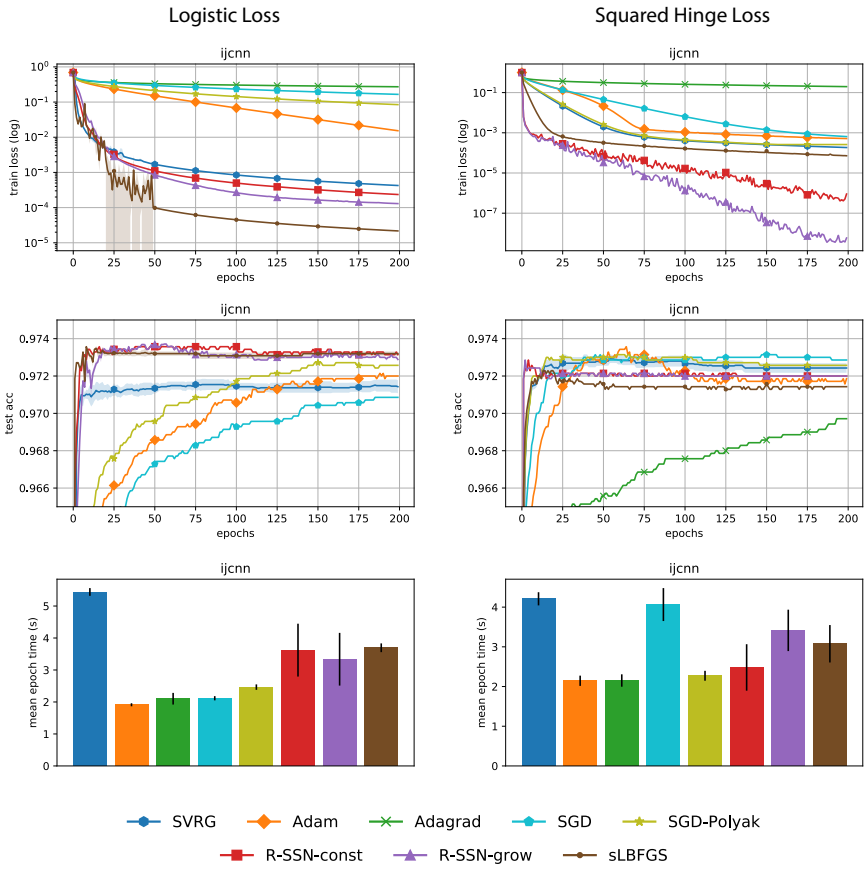

Figure 5.3: Comparison of R-SSN variants and stochastic L-BFGS against first-order methods on the `ijcnn` dataset. Although the interpolation condition is not satisfied, higher-order methods are still competitive.

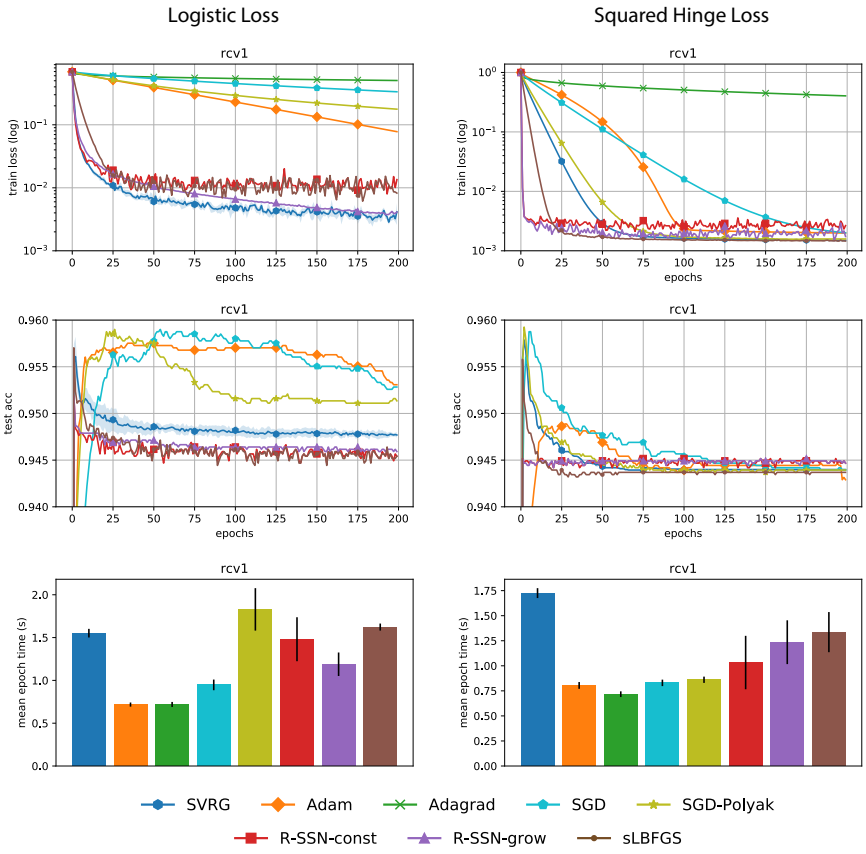

Figure 5.4: Comparison of R-SSN variants and stochastic L-BFGS against first order methods on the `rcv1` dataset. Although the interpolation condition is not satisfied, higher-order methods are still competitive.

Chapter 6

Conclusion

We showed that the regularized subsampled Newton's method (R-SSN) method with a constant batch-size achieves linear convergence rates when minimizing smooth functions that are strongly-convex or self-concordant under the interpolation setting. We also showed that interpolation enables stochastic BFGS-type methods to converge linearly. We validated our theoretical claims via experiments using kernel functions and demonstrated the fast convergence of the proposed methods. Our theoretical and empirical results show the potential for training large over-parameterized models that satisfy the interpolation property. For future work, we aim to investigate ways to handle non-convex losses and explore practical line-search strategies so that stochastic second-order methods can be scaled to optimize over millions of parameters.

Bibliography

Naman Agarwal, Brian Bullins, and Elad Hazan. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, 18(1):4148–4187, 2017. → page 12

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 2019. → page 5

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. → page 13

Francis Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010. → pages 21, 22

Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. *arXiv preprint:1811.02564*, 2018. → pages 6, 11

Sue Becker and Yann Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 Connectionist Models Summer School*, 1988. → page 13

Mikhail Belkin, Alexander Rakhlin, and Alexandre B. Tsybakov. Does data interpolation contradict statistical optimality? In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2019. → page 5

Stefania Bellavia, Nataša Krejić, and Nataša Krklec Jerinkić. Subsampled inexact Newton methods for minimizing large sums of convex functions. *IMA Journal of Numerical Analysis*, 2018. → pages 6, 13

Albert S. Berahas, Majid Jahani, and Martin Takáč. Quasi-Newton methods for deep learning: Forget the past, just sample. *arXiv preprint:1901.09997*, 2019. → page 27

Albert S. Berahas, Raghu Bollapragada, and Jorge Nocedal. An investigation of Newton-sketch and subsampled Newton methods. *Optimization Methods and Software*, pages 1–20, 2020. → page 15

El-houcine Bergou, Youssef Diouane, Vyacheslav Kungurtsev, and Clément W. Royer. A subsampling line-search method with second-order results. *arXiv preprint:1810.07211*, 2018. → page 13

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific, 2016. → page 6

Raghu Bollapragada, Richard H. Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 2018a. → pages 4, 6, 7, 13, 14, 15, 17, 52

Raghu Bollapragada, Dheevatsa Mudigere, Jorge Nocedal, Hao-Jun Michael Shi, and Ping Tak Peter Tang. A progressive batching L-BFGS method for machine learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018b. → pages 7, 28, 29, 30, 31

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. → pages 19, 20, 21

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. → page 12

Richard H. Byrd, Gillian M. Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012. → pages 13, 28

Eric Carlen. Trace inequalities and quantum entropy: an introductory course. *Entropy and the Quantum*, 529:73–140, 2010. → page 71

Augustin Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *C.R. Acad Sci Par*, 25(1847):536–538, 1847. → page 1

Volkan Cevher and Bang Công Vũ. On the linear convergence of the stochastic gradient method with constant step-size. *Optimization Letters*, 13(5): 1177–1187, 2019. → page 6

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 1–27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. → page 35

Daqing Chang, Shiliang Sun, and Changshui Zhang. An accelerated linearly convergent stochastic L-BFGS algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3338–3346, 2019. → page 28

Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Big batch SGD: Automated inference using adaptive batch sizes. *arXiv preprint:1610.05792*, 2016. → page 16

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, NeurIPS*, 2014. → page 4

Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19:400–408, 1982. → page 16

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. → page 27

Simon S. Du, Xiyu Zhai, Barnabás Póczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *7th International Conference on Learning Representations, ICLR*, 2019. → page 5

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011. → pages 4, 7, 26, 32

Mark Eisen, Aryan Mokhtari, and Alejandro Ribeiro. Large scale empirical risk minimization via truncated adaptive Newton method. In *International Conference on Artificial Intelligence and Statistics, AISTATS*, 2018. → page 21

Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems, NeurIPS*, 2015. → pages 4, 12

Michael P. Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3): A1380–A1405, 2012. → pages 16, 28, 31

Wenbo Gao and Donald Goldfarb. Quasi-Newton methods: superlinear convergence without line searches for self-concordant functions. *Optimization Methods and Software*, 34(1):194–217, 2019. → page 29

Robert M. Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: squeezing more curvature out of data. In *Proceedings of the 33rd International Conference on Machine Learning, ICML*, 2016. → page 28

Robert M. Gower, Filip Hanzely, Peter Richtárik, and Sebastian U. Stich. Accelerated stochastic matrix inversion: General theory and speeding up BFGS rules for faster second-order optimization. In *Advances in Neural Information Processing Systems, NeurIPS*, 2018. → page 6

Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečnỳ, and Scott Sallinen. Stop wasting my gradients: Practical SVRG. In *Advances in Neural Information Processing Systems, NeurIPS*, 2015. → pages 15, 23, 60

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. → pages 2, 7, 31

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems, NeurIPS*, 2013. → pages 4, 7, 32

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD*, 2016. → page 30

Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of Newton's method without strong-convexity or Lipschitz gradients. *arXiv preprint:1806.00413*, 2018. → page 14

Carl T. Kelley. A brief introduction to implicit filtering. Technical Report CRSC-TR02-28, North Carolina State University. Center for Research in Scientific Computation, 2002. URL https://repository.lib.ncsu.edu/handle/1840.4/550. → page 29

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015. → pages 4, 7, 32

Ritesh Kolte, Murat Erdogdu, and Ayfer Ozgur. Accelerating SVRG via second-order information. In *NIPS Workshop on Optimization for Machine Learning*, 2015. → pages 7, 28

Dmitry Kovalev, Robert M. Gower, Peter Richtárik, and Alexander Rogozin. Fast linear convergence of randomized BFGS. *arXiv preprint:2002.11337*, 2020. → page 30

Nicolas Le Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *Advances in Neural Information Processing Systems, NeurIPS*, 2007. → page 13

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944. → pages 6, 10

Xiang Li, Shusen Wang, and Zhihua Zhang. Do subsampled Newton methods work for high-dimensional data? In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, 2020. → page 12

Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "ridgeless" regression can generalize. *arXiv preprint:1808.00387*, 2018. → page 5

Chaoyue Liu and Mikhail Belkin. MaSS: an accelerated stochastic method for over-parametrized learning. *arXiv preprint:1810.13395*, 2018. → page 6

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. → pages 7, 27

Jie Liu, Yu Rong, Martin Takác, and Junzhou Huang. On the acceleration of L-BFGS with second-order information and stochastic batches. *arXiv preprint:1807.05328*, 2018. → page 28

Sharon L. Lohr. *Sampling: Design and Analysis*. Chapman and Hall/CRC, 2019. → pages 49, 60

Aurelien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic Newton method. *arXiv preprint:1503.08316*, 2015. → pages 7, 28, 29, 30

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018. → pages 6, 11

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11 (2):431–441, 1963. → pages 6, 10

Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Globally convergent Newton methods for ill-conditioned generalized self-concordant losses. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019a. → pages 7, 22

Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Conference on Learning Theory, COLT*, 2019b. → page 21

James Martens. Deep learning via Hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning, ICML*, 2010. → page 10

James Martens and Roger B. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015. → page 13

Si Yi Meng, Sharan Vaswani, Issam Laradji, Mark Schmidt, and Simon Lacoste-Julien. Fast and furious convergence: Stochastic second order methods under interpolation. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2020. → page v

Andre Milzarek, Xiantao Xiao, Shicong Cen, Zaiwen Wen, and Michael Ulbrich. A stochastic semismooth Newton method for nonsmooth nonconvex optimization. *SIAM Journal on Optimization*, 29(4):2916–2948, 2019. → page 13

Aryan Mokhtari, Hadi Daneshmand, Aurélien Lucchi, Thomas Hofmann, and Alejandro Ribeiro. Adaptive Newton method for empirical risk minimization to statistical accuracy. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016. → page 21

Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. IQN: An incremental quasi-Newton method with local superlinear convergence rate. *SIAM Journal on Optimization*, 28(2):1670–1698, 2018. → page 29

Philipp Moritz, Robert Nishihara, and Michael I. Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2016. → pages 7, 28, 29, 30

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018. → pages 2, 11, 15, 16, 66

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994. → pages 20, 21

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Science & Business Media, 2006. → pages 9, 10, 27, 76

James M. Ortega and Werner C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. SIAM, 1970. → page 9

Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. In *2nd International Conference on Learning Representations, ICLR*, 2014. → page 13

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019. → page 32

Vivak Patel. Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning. *SIAM Journal on Optimization*, 26(4):2620–2648, 2016. → page 13

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994. → page 2

Mert Pilanci and Martin J. Wainwright. Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. *SIAM Journal on Optimization*, 27(1):205–245, 2017. → pages 12, 21

Boris T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963. → page 30

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5): 1–17, 1964. → page 32

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951. → page 3

Anton Rodomanov and Dmitry Kropotov. A superlinearly-convergent proximal Newton-type method for the optimization of finite sums. In *Proceedings of the 33rd International Conference on Machine Learning, ICML*, 2016. → page 28

Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods I: globally convergent algorithms. *arXiv preprint:1601.04737*, 2016a. → pages 4, 12

Farbod Roosta-Khorasani and Michael W. Mahoney. Sub-sampled Newton methods II: Local convergence rates. *arXiv preprint:1601.04738*, 2016b. → pages 4, 12, 15

Robert E. Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998. → page 5

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint:1308.6370*, 2013. → pages 5, 6, 12, 14

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017. → page 4

Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS*, 2007. → pages 28, 29

Peter Sunehag, Jochen Trumpf, S. V. N. Vishwanathan, and Nicol N. Schraudolph. Variable metric stochastic approximation theory. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2009. → pages 28, 29

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012. → page 4

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. → page 71

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *The*

22nd International Conference on Artificial Intelligence and Statistics, AISTATS, 2019a. → pages 6, 11, 12, 14

Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019b. → pages 6, 8, 11, 31, 32, 35

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, NeurIPS*, 2016. → pages 4, 12

Haishan Ye, Luo Luo, and Zhihua Zhang. Approximate Newton methods and their local convergence. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, 2017. → pages 12, 15, 16, 17

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR*, 2017. → page 5

Yuchen Zhang and Xiao Lin. DiSCO: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015. → pages 21, 24, 71

Renbo Zhao, William B. Haskell, and Vincent Y. F. Tan. Stochastic L-BFGS: Improved convergence rates and practical acceleration strategies. *IEEE Transactions on Signal Processing*, 66(5):1155–1169, 2018. → page 28

Chaoxu Zhou, Wenbo Gao, and Donald Goldfarb. Stochastic Adaptive Quasi-Newton Methods for Minimizing Expected Values. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, 2017. → page 29

Appendix A

Supporting Materials

A.1 Common results

Lemma 1. *Consider* $y = \frac{1}{n}\sum_{i=1}^{n} y_i$ *where* $y_i \in \mathbb{R}^d$. *Then for a* y_i *selected uniformly at random, we have* $\mathbb{E}[y_i] = y$. *Suppose we uniformly draw a sample* $B \subset \{1, \ldots, n\}$ *and let* $y_B = \frac{1}{b}\sum_{i \in B}$ *where* $b = |B|$. *If the* y_i*'s satisfy a growth condition such that*

$$\mathbb{E}_i \|y_i\|^2 \leq c \|y\|^2$$

for some $c > 0$. *Then the expected squared norm of the error* $\epsilon = y_B - y$ *can be bounded as*

$$\mathbb{E}\|y_B - y\|^2 \leq \frac{(n-b)(c-1)}{(n-1)b}\|y\|^2.$$

Proof. For an arbitrary entry $j \in \{1, \ldots, d\}$, the error ϵ_j^2 can be bounded using its sample variance [Lohr, 2019] as follows,

$$\begin{aligned}\epsilon_j^2 &= \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}((y_i)_j - y_j)^2 \\ &= \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}((y_i)_j^2 - 2(y_i)_j y_j + y_j^2).\end{aligned}$$

Take the squared norm of ϵ, we have

$$\begin{aligned}\|\epsilon\|^2 &= \frac{n-b}{nb}\frac{1}{n-1}\sum_{j=1}^{d}\sum_{i=1}^{n}((y_i)_j^2 - 2(y_i)_j y_j + y_j^2) \\ &= \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}\left(\|y_i\|^2 - 2\langle y_i\,,\, y\rangle + \|y\|^2\right).\end{aligned}$$

Now take expectation on both sides and use the unbiasedness of y_i gives us

$$\mathbb{E}\,\|\epsilon\|^2 = \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}\left(\mathbb{E}\,\|y_i\|^2 - 2\,\|y\|^2 + \|y\|^2\right). \tag{A.1}$$

Apply the growth condition,

$$\begin{aligned}\mathbb{E}\,\|\epsilon\|^2 &\le \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}\left(c\,\|y\|^2 - \|y\|^2\right) \\ &= \frac{(n-b)(c-1)}{(n-1)b}\,\|y\|^2\end{aligned}$$

which completes the proof.

□

Lemma 2. *Consider the same setup as in Lemma 1. If we replace the growth condition with*

$$\mathbb{E}_i\,\|y_i\|^2 \le c\,\|y\|^4,$$

then we obtain the following bound on the expected squared error

$$\mathbb{E}\,\|y_B - y\|^2 \le \frac{(n-b)c}{(n-1)b}\,\|y\|^4$$

for some $c > 0$.

Proof. Using the same analysis as in Lemma 1 up to equation (A.1) and applying

the new growth condition gives us

$$\begin{aligned}
\mathbb{E}\left\|\epsilon\right\|^2 &\leq \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}\left(c\left\|y\right\|^4 - \left\|y\right\|^2\right) \\
&\leq \frac{n-b}{nb}\frac{1}{n-1}\sum_{i=1}^{n}\left(c\left\|y\right\|^4\right) && \text{(Since } \left\|y\right\|^2 > 0\text{)} \\
&= \frac{(n-b)c}{(n-1)b}\left\|y\right\|^4.
\end{aligned}$$

□

A.2 Proof of Theorem 1

We restate Theorem 1.

Theorem 1 (Global linear convergence)**.** *Under μ-strong convexity, L-smoothness, and ρ-*SGC*, the sequence $\{w_k\}_{k\geq 0}$ generated by* R-SSN *with step size*

$$\eta_k = \frac{(\mu_{\mathcal{S}_k} + \tau)^2}{L\left((\mu_{\mathcal{S}_k} + \tau) + (L_{\mathcal{S}_k} + \tau)\, c_g\right)} \tag{2.11}$$

and constant batch sizes $b_{s_k} = b_s$, $b_{g_k} = b_g$ converges to w^* at a Q-linear rate from an arbitrary initialization w_0,

$$\mathbb{E}[f(w_T)] - f(w^*) \leq (1-\alpha)^T \left(f(w_0) - f(w^*)\right), \tag{2.12}$$

where the constants are given by

$$\alpha = \min\left\{\frac{(\bar{\mu}+\tau)^2}{2\kappa c_g(\tilde{L}+\tau)}\,,\, \frac{(\bar{\mu}+\tau)}{2\kappa(\tilde{L}+\tau)}\right\} \quad \textit{and} \quad c_g = \frac{(\rho-1)\,(n-b_g)}{(n-1)\,b_g}. \tag{2.13}$$

Proof. This analysis closely follows the proof of Theorem 2.2 in Bollapragada et al. [2018a]. By the L-smoothness assumption,

$$\begin{aligned} f(w_{k+1}) &\leq f(w_k) + \langle \nabla f(w_k)\,,\, w_{k+1} - w_k\rangle + \frac{L}{2}\left\|w_{k+1} - w_k\right\|^2 \\ &= f(w_k) - \eta_k \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f_{\mathcal{G}_k}(w_k)\right\rangle \\ &\qquad + \frac{L}{2}\eta_k^2 \left\|[\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f_{\mathcal{G}_k}(w_k)\right\|^2. \qquad \text{(Update step)} \end{aligned}$$

Since $\mathcal{S}_k$ and $\mathcal{G}_k$ are independent samples, we can fix the Hessian sample $\mathcal{S}_k$ and take expectation with respect to the unbiased gradient sample $\mathcal{G}_k$,

$$\begin{aligned} \mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] &\leq f(w_k) - \eta_k \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f(w_k)\right\rangle \\ &\qquad + \frac{L}{2}\eta_k^2 \underbrace{\mathbb{E}_{\mathcal{G}_k}\left\|[\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f_{\mathcal{G}_k}(w_k)\right\|^2}_{:=P}. \end{aligned}$$

Using the fact $\mathbb{E}\left\|x\right\|^2 = \mathbb{E}\left\|x - \mathbb{E}x\right\|^2 + \left\|\mathbb{E}x\right\|^2$ with $x = [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1}\nabla f_{\mathcal{G}_k}(w_k)$,

we can bound the last term as

$$
\begin{aligned}
P &= \mathbb{E}_{\mathcal{G}_k}\left[\left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f_{\mathcal{G}_k}(w_k) - \mathbb{E}_{\mathcal{G}_k}\left[\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f_{\mathcal{G}_k}(w_k)\right]\right\|^2\right] \\
&\quad + \left\|\mathbb{E}_{\mathcal{G}_k}\left[\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f_{\mathcal{G}_k}(w_k)\right]\right\|^2 \\
&= \mathbb{E}_{\mathcal{G}_k}\left[\left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\left[\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\right]\right\|^2\right] \\
&\quad + \left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f(w_k)\right\|^2. \\
&\qquad\qquad\qquad\qquad \text{(Again, by unbiasedness and independent batches)} \\
&\leq \frac{1}{\left(\mu_{\mathcal{S}_k}+\tau\right)^2}\mathbb{E}_{\mathcal{G}_k}\left[\left\|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\right\|^2\right] \\
&\quad + \left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f(w_k)\right\|^2. \qquad\qquad \text{(Since } \mathbf{H}_{\mathcal{S}_k}(w_k) \succeq (\mu_{\mathcal{S}_k}+\tau)I_d\text{)}
\end{aligned}
$$

Now we use Lemma 1 in Appendix A.1 to obtain

$$
\mathbb{E}_{\mathcal{G}_k}\left\|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\right\|^2 \leq \frac{n-b_{g_k}}{(n-1)\,b_{g_k}}(\rho-1)\left\|\nabla f(w_k)\right\|^2,
$$

which implies

$$
P \leq \frac{\rho-1}{\left(\mu_{\mathcal{S}_k}+\tau\right)^2}\frac{n-b_{g_k}}{(n-1)\,b_{g_k}}\left\|\nabla f(w_k)\right\|^2 + \left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f(w_k)\right\|^2.
$$

From the above relations, we have

$$
\begin{aligned}
\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] &\leq f(w_k) + \frac{L\eta_k^2}{2}\frac{\rho-1}{\left(\mu_{\mathcal{S}_k}+\tau\right)^2}\frac{n-b_{g_k}}{(n-1)\,b_{g_k}}\left\|\nabla f(w_k)\right\|^2 \\
&\quad \underbrace{-\eta_k\left\langle\nabla f(w_k)\,,\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f(w_k)\right\rangle + \frac{L\eta_k^2}{2}\left\|\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\nabla f(w_k)\right\|^2}_{:=Q}.
\end{aligned}
$$

Expanding $\left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f(w_k) \right\|^2$ and decomposing $\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}$ gives us

$$\begin{aligned}
Q &= -\eta_k \left\langle \nabla f(w_k)\,,\, \left(\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} - \frac{L\eta_k}{2} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-2} \right) \nabla f(w_k) \right\rangle \\
&= -\eta_k \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \nabla f(w_k) \right\|^2_{\left(I - \frac{L\eta_k}{2}\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\right)} \\
&\leq -\eta_k \left(1 - \frac{L\eta_k}{2\left(\mu_{\mathcal{S}_k} + \tau\right)} \right) \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \nabla f(w_k) \right\|^2 \\
&\leq -\frac{\eta_k}{\left(L_{\mathcal{S}_k} + \tau\right)} \left(1 - \frac{L\eta_k}{2\left(\mu_{\mathcal{S}_k} + \tau\right)} \right) \left\| \nabla f(w_k) \right\|^2,
\end{aligned}$$

where the second last inequality requires $\left(I - \frac{L\eta_k}{2}\left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1}\right)$ to be positive definite, which is true for step-sizes satisfying

$$\eta_k \leq \frac{2\left(\mu_{\mathcal{S}_k} + \tau\right)}{L}. \tag{A.2}$$

Then we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] &\leq f(w_k) - \frac{\eta_k}{\left(L_{\mathcal{S}_k} + \tau\right)} \left(1 - \frac{L\eta_k}{2\left(\mu_{\mathcal{S}_k} + \tau\right)} \right) \left\| \nabla f(w_k) \right\|^2 \\
&\qquad + \frac{L\eta_k^2}{2} \frac{\rho - 1}{\left(\mu_{\mathcal{S}_k} + \tau\right)^2} \frac{n - b_{g_k}}{(n-1)\, b_{g_k}} \left\| \nabla f(w_k) \right\|^2 \\
&= f(w_k) - \left[\frac{\eta_k}{\left(L_{\mathcal{S}_k} + \tau\right)} \left(1 - \frac{L\eta_k}{2\left(\mu_{\mathcal{S}_k} + \tau\right)} \right) \right. \\
&\qquad \left. - \frac{L\eta_k^2}{2} \frac{\rho - 1}{\left(\mu_{\mathcal{S}_k} + \tau\right)^2} \frac{n - b_{g_k}}{(n-1)\, b_{g_k}} \right] \left\| \nabla f(w_k) \right\|^2.
\end{aligned}$$

Subtracting $f(w^*)$ from both sides and using the fact that strong convexity implies

$\|\nabla f(x)\|^2 \geq 2\mu(f(x) - f(x^*))$ for all x, the above bound becomes

$$\begin{aligned}\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) &\leq f(w_k) - f(w^*) - 2\mu\left[\frac{\eta_k}{(L_{\mathcal{S}_k}+\tau)}\left(1 - \frac{L\eta_k}{2\,(\mu_{\mathcal{S}_k}+\tau)}\right)\right. \\ &\qquad \left. - \frac{L\eta_k^2}{2}\frac{\rho-1}{(\mu_{\mathcal{S}_k}+\tau)^2}\frac{n-b_{g_k}}{(n-1)\,b_{g_k}}\right](f(w_k) - f(w^*)) \\ &\leq \left(1 - \frac{2\mu\eta_k}{(L_{\mathcal{S}_k}+\tau)} + \frac{\mu L\eta_k^2}{(L_{\mathcal{S}_k}+\tau)\,(\mu_{\mathcal{S}_k}+\tau)}\right. \\ &\qquad \left. + \frac{\mu L(\rho-1)\eta_k^2}{(\mu_{\mathcal{S}_k}+\tau)^2}\frac{n-b_{g_k}}{(n-1)\,b_{g_k}}\right)(f(w_k) - f(w^*))\end{aligned}$$

Define constants $c_1 = \frac{\mu}{(L_{\mathcal{S}_k}+\tau)}$, $c_2 = \frac{L}{(\mu_{\mathcal{S}_k}+\tau)}$, and $c_3 = \frac{\mu(\rho-1)}{(\mu_{\mathcal{S}_k}+\tau)}\frac{n-b_{g_k}}{(n-1)\,b_{g_k}}$ using constant batch-sizes for the subsampled gradient and Hessian, i.e. $b_{g_k} = b_g \geq 1$ and $b_{s_k} = b_s \geq 1$, we have

$$\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) \leq \left(1 - 2c_1\eta_k + c_1c_2\eta_k^2 + c_2c_3\eta_k^2\right)(f(w_k) - f(w^*)).$$

To ensure contraction, the step size needs to satisfy

$$\left(1 - 2c_1\eta_k + c_1c_2\eta_k^2 + c_2c_3\eta_k^2\right) \in (0, 1],$$

which gives

$$0 < \eta_k \leq \frac{2c_1}{c_2(c_1 + c_3)}. \tag{A.3}$$

Taking $\eta_k \leq \frac{c_1}{c_2(c_1+c_3)}$, the above bound becomes

$$\begin{aligned}\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) &\leq \left(1 - \frac{c_1^2}{c_2(c_1+c_3)}\right)(f(w_k) - f(w^*)) \\ &\leq \left(1 - \underbrace{\frac{\mu\ (\mu_{\mathcal{S}_k}+\tau)^2}{L\ (L_{\mathcal{S}_k}+\tau)\ (\mu_{\mathcal{S}_k}+\tau) + c_gL\ (L_{\mathcal{S}_k}+\tau)}}_{:=C}\right)(f(w_k) - f(w^*)),\end{aligned}$$

where we denote $c_g = \frac{(\rho-1)(n-b_g)}{(n-1)\,b_g}$. Note that since $c_g \geq 0$, our bound for η_k

simplifies to

$$\eta_k \leq \frac{\left(\mu_{\mathcal{S}_k}+\tau\right)^2}{L\left(\left(\mu_{\mathcal{S}_k}+\tau\right)+\left(L_{\mathcal{S}_k}+\tau\right)c_g\right)} \leq \frac{\left(\mu_{\mathcal{S}_k}+\tau\right)^2}{L\left(\mu_{\mathcal{S}_k}+\tau\right)} \leq \frac{\left(\mu_{\mathcal{S}_k}+\tau\right)}{L},$$

hence satisfies the earlier requirement in (A.2). Now we lower bound the term C,

$$\begin{aligned} C &\geq \frac{\mu\ \left(\mu_{\mathcal{S}_k}+\tau\right)^2}{\left(\tilde{L}+\tau\right)\ L\ \left(\mu_{\mathcal{S}_k}+\tau+c_g\right)} \\ &\geq \frac{\mu\ \left(\mu_{\mathcal{S}_k}+\tau\right)^2}{2\max\{c_g,\left(\mu_{\mathcal{S}_k}+\tau\right)\}\ \left(\tilde{L}+\tau\right)\ L}, \qquad \left(\tfrac{a+b}{2} \leq \max\{a,b\}\right) \end{aligned}$$

which gives

$$\begin{aligned} &\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) \\ &\quad \leq \left(1 - \frac{\mu\ \left(\mu_{\mathcal{S}_k}+\tau\right)^2}{2\max\{c_g,\left(\mu_{\mathcal{S}_k}+\tau\right)\}\ \left(\tilde{L}+\tau\right)\ L}\right)\left(f(w_k) - f(w^*)\right). \end{aligned}$$

Case 1: If $c_g \geq \left(\mu_{\mathcal{S}_k}+\tau\right)$, then

$$\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) \leq \left(1 - \frac{\left(\mu_{\mathcal{S}_k}+\tau\right)^2\ \mu}{2c_g\ \left(\tilde{L}+\tau\right)\ L}\right)\left(f(w_k) - f(w^*)\right).$$

Taking an expectation w.r.t. $\mathcal{S}_k$ yields

$$\begin{aligned} \mathbb{E}_{\mathcal{S}_k,\mathcal{G}_k}[f(w_{k+1})] - f(w^*) &\leq \mathbb{E}_{\mathcal{S}_k}\left(1 - \frac{\left(\mu_{\mathcal{S}_k}+\tau\right)^2\ \mu}{2c_g\ \left(\tilde{L}+\tau\right)\ L}\right)\left(f(w_k) - f(w^*)\right) \\ &\leq \left(1 - \frac{\left(\mathbb{E}_{\mathcal{S}_k}[\mu_{\mathcal{S}_k}]+\tau\right)^2\ \mu}{2c_g\ \left(\tilde{L}+\tau\right)\ L}\right)\left(f(w_k) - f(w^*)\right) \\ &\qquad \text{(By Jensen's inequality)} \\ &\leq \left(1 - \frac{\left(\bar{\mu}+\tau\right)^2\ \mu}{2c_g\ \left(\tilde{L}+\tau\right)\ L}\right)\left(f(w_k) - f(w^*)\right). \end{aligned}$$

Case 2: If $c_g \leq (\mu_S + \tau)$, then

$$\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) \leq \left(1 - \frac{\mu\ (\mu_{\mathcal{S}_k} + \tau)}{2\left(\tilde{L} + \tau\right)\ L}\right)(f(w_k) - f(w^*)).$$

Taking an expectation w.r.t. $\mathcal{S}_k$ yields

$$\mathbb{E}_{\mathcal{S}_k, \mathcal{G}_k}[f(w_{k+1})] - f(w^*) \leq \left(1 - \frac{\mu(\bar{\mu} + \tau)}{2\left(\tilde{L} + \tau\right)\ L}\right)(f(w_k) - f(w^*)).$$

Putting the two cases together, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}_k, \mathcal{G}_k}[f(w_{k+1})] - f(w^*) \\
&\leq \max\left\{\left(1 - \frac{(\bar{\mu} + \tau)^2\ \mu}{2c_g\left(\tilde{L} + \tau\right)\ L}\right), \left(1 - \frac{\mu(\bar{\mu} + \tau)}{2\left(\tilde{L} + \tau\right)\ L}\right)\right\}(f(w_k) - f(w^*)).
\end{aligned}$$

Now we take expectation over all time steps, apply recursion, and simplify using the definition for the condition number $\kappa = \frac{L}{\mu}$

$$\begin{aligned}
&\mathbb{E}[f(w_T)] - f(w^*) \\
&\leq \left(1 - \min\left\{\frac{(\bar{\mu} + \tau)^2}{2\kappa c_g\left(\tilde{L} + \tau\right)}, \frac{(\bar{\mu} + \tau)}{2\kappa\left(\tilde{L} + \tau\right)}\right\}\right)^T (f(w_0) - f(w^*))
\end{aligned}$$

and the proof is complete.

□

A.3 Proof of Theorem 2

We restate Theorem 2.

Theorem 2 (Local convergence)**.** *Suppose Assumptions 1 to 3 in Theorem 1 are satisfied. Additionally, under the M-Lipschitz continuous Hessian, γ-bounded moments of the iterates, and σ^2-bounded variance of the subsampled Hessian assumptions, the sequence $\{w_k\}_{k\geq 0}$ generated by* R-SSN *with unit step-size $\eta_k = 1$ and growing batch-sizes satisfying*

$$b_{g_k} \geq \frac{n}{(\frac{n-1}{\rho-1})\left\|\nabla f(w_k)\right\|^2 + 1}, \quad b_{s_k} \geq \frac{n}{\frac{n}{\sigma^2}\left\|\nabla f(w_k)\right\| + 1} \tag{2.19}$$

converges to w^* at a linear-quadratic rate

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \frac{\gamma(M + 2L + 2L^2)}{2(\tilde{\mu} + \tau)}\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2 + \frac{\tau}{\tilde{\mu} + \tau}\mathbb{E}\left\|w_k - w^*\right\| \tag{2.20}$$

in a local neighbourhood of the solution where $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau)}{\gamma(M+2L+2L^2)}$. Furthermore, when $\tau = 0$ and $\tilde{\mu} > 0$, R-SSN *can achieve local quadratic convergence*

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \gamma\Big(\frac{M + 2L + 2L^2}{2\tilde{\mu}}\Big)\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2.$$

Proof. From the update rule,

$$
\begin{aligned}
&\|w_{k+1} - w^*\| \\
&= \left\| w_k - w^* - \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\| \\
&= \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \left(\mathbf{H}_{\mathcal{S}_k}(w_k)(w_k - w^*) - \nabla f_{\mathcal{G}_k}(w_k)\right) \right\| \\
&= \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \left(\mathbf{H}_{\mathcal{S}_k}(w_k)(w_k - w^*) - \nabla f(w_k) - \nabla f_{\mathcal{G}_k}(w_k) + \nabla f(w_k)\right) \right\| \\
&\leq \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \right\| \left\| \mathbf{H}_{\mathcal{S}_k}(w_k)(w_k - w^*) - \nabla f(w_k) - \nabla f_{\mathcal{G}_k}(w_k) + \nabla f(w_k) \right\| \\
&\leq \frac{1}{(\mu_{S_k} + \tau)} \left\| \mathbf{H}_{\mathcal{S}_k}(w_k)(w_k - w^*) - \nabla f(w_k) - \nabla f_{\mathcal{G}_k}(w_k) + \nabla f(w_k) \right\| \\
&= \frac{1}{(\mu_{S_k} + \tau)} \Big\| \mathbf{H}_{\mathcal{S}_k}(w_k)(w_k - w^*) - \nabla^2 f(w_k)(w_k - w^*) \\
&\qquad\qquad + \nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k) - \nabla f_{\mathcal{G}_k}(w_k) + \nabla f(w_k) \Big\|
\end{aligned}
$$

where we repeatedly applied the triangle inequality. Then we have

$$
\begin{aligned}
\|w_{k+1} - w^*\| \leq \frac{1}{(\mu_{S_k} + \tau)} \Big[& \left\| \nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k) \right\| \\
& + \left\| \left(\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right)(w_k - w^*) \right\| \\
& + \left\| \nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k) \right\| \Big].
\end{aligned}
$$

Taking the expectation $\mathbb{E}_k$ over all combinations of $\mathcal{S}_k$ and $\mathcal{G}_k$, we have

$$
\begin{aligned}
\mathbb{E}_k \|w_{k+1} - w^*\| \leq \frac{1}{(\tilde{\mu} + \tau)} \Big[& \underbrace{\left\| \nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k) \right\|}_{\text{(1) Bound using Lipschitz Hessian}} \\
& + \underbrace{\mathbb{E}_k \left\| \left(\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right)(w_k - w^*) \right\|}_{\text{(2) Bound using Hessian Variance}} \\
& + \underbrace{\mathbb{E}_k \left\| \nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k) \right\|}_{\text{(3) Bound using SGC}} \Big].
\end{aligned}
$$

We bound the first term using M-Lipschitz continuity of the Hessian,

$$
\begin{aligned}
&\left\|\nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k)\right\| \\
&= \left\|\nabla f(w_k) - \nabla f(w^*) - \nabla^2 f(w_k)(w_k - w^*)\right\| \\
&= \left\|\int_0^1 \nabla^2 f(w^* + t(w_k - w^*))(w_k - w^*)dt - \nabla^2 f(w_k)(w_k - w^*)\right\| \\
&= \left\|\int_0^1 \left(\nabla^2 f(w^* + t(w_k - w^*) - \nabla^2 f(w_k)\right)(w_k - w^*)dt\right\| \\
&\leq \int_0^1 \left\|\left(\nabla^2 f(w^* + t(w_k - w^*) - \nabla^2 f(w_k)\right)(w_k - w^*)dt\right\| \\
&\hspace{20em} \text{(Jensen's inequality)} \\
&\leq \int_0^1 \left\|\nabla^2 f(w^* + t(w_k - w^*)) - \nabla^2 f(w_k)\right\| \left\|w_k - w^*\right\| dt \\
&\hspace{20em} \text{(Cauchy–Schwarz inequality)} \\
&\leq \int_0^1 M \left\|w^* + t(w_k - w^*) - w_k\right\| \left\|w_k - w^*\right\| dt \qquad (M\text{-Lipschitz Hessian}) \\
&= \left\|w_k - w^*\right\|^2 \int_0^1 M\,(1-t)\,dt,
\end{aligned}
$$

giving us

$$
\left\|\nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k)\right\| \leq \frac{M}{2} \left\|w_k - w^*\right\|^2.
$$

To bound the second term, we use

$$
\begin{aligned}
\mathbb{E}_k \left\|\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right\| &= \mathbb{E}_k \left\|\nabla^2 f_{\mathcal{S}_k}(w_k) + \tau I - \nabla^2 f(w_k)\right\| \\
&\leq \mathbb{E}_k \left\|\nabla^2 f_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right\| + \left\|\tau I\right\|.
\end{aligned}
$$

By the assumption that the subsampled Hessians have bounded variance, from Harikandeh et al. [2015], Lohr [2019] we have that

$$
\mathbb{E}_k \left[\left\|\nabla^2 f_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right\|^2\right] \leq \frac{n - b_{s_k}}{n\, b_{s_k}} \sigma_s^2.
$$

Using Jensen's inequality for the square root function and combining with the above, we have

$$\mathbb{E}_k \left\| \mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k) \right\| \leq \sigma_s \sqrt{\frac{n - b_{s_k}}{n\, b_{s_k}}} + \tau.$$

Then by setting the subsampled Hessian batch-size according to

$$b_{s_k} \geq \frac{n}{\|\nabla f(w_k)\| \frac{n}{\sigma_s^2} + 1},$$

we can achieve

$$\mathbb{E}_k \left\| \mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k) \right\| \leq \|\nabla f(w_k)\| + \tau.$$

The second term can then be bounded as

$$\begin{aligned}
&\mathbb{E}_k \left\| \left(\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k) \right) (w_k - w^*) \right\| \\
&\quad \leq \|w_k - w^*\| \, \mathbb{E}_k \left\| \mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k) \right\| && \text{(Cauchy-Schwarz)} \\
&\quad \leq \|w_k - w^*\| \, (\|\nabla f(w_k)\| + \tau) && \text{(from above)} \\
&\quad \leq L \|w_k - w^*\|^2 + \tau \|w_k - w^*\|. && (L\text{-smoothness})
\end{aligned}$$

The third term can again be bounded using Lemma 1. Applying Jensen's inequality, this gives us

$$\mathbb{E}_k \|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\| \leq \sqrt{\frac{n - b_{g_k}}{(n-1)\, b_{g_k}}} \sqrt{\rho - 1} \, \|\nabla f(w_k)\|.$$

If we let $b_{g_k} \geq \frac{n}{\left(\frac{n-1}{\rho-1}\right)\|\nabla f(w_k)\|^2 + 1}$, then we have

$$\mathbb{E}_k \|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\| \leq \|\nabla f(w_k)\|^2 \leq L^2 \|w_k - w^*\|^2,$$

where the last inequality again comes from L-smoothness. Putting the above three

bounds together gives us

$$\begin{aligned}
&\mathbb{E}_k \left\| w_{k+1} - w^* \right\| \\
&\leq \frac{1}{(\tilde{\mu} + \tau)} \left[\frac{M}{2} \left\| w_k - w^* \right\|^2 + L \left\| w_k - w^* \right\|^2 \right. \\
&\qquad\qquad \left. + \tau \left\| w_k - w^* \right\| + L^2 \left\| w_k - w^* \right\|^2 \right] \\
&\leq \frac{\left(M + 2L + 2L^2 \right)}{2(\tilde{\mu} + \tau)} \left\| w_k - w^* \right\|^2 + \frac{\tau}{\tilde{\mu} + \tau} \left\| w_k - w^* \right\|.
\end{aligned}$$

Using the γ bounded moments assumption by taking expectation over all k yields

$$\mathbb{E} \left\| w_{k+1} - w^* \right\| \leq \frac{\gamma \left(M + 2L + 2L^2 \right)}{2(\tilde{\mu} + \tau)} \left(\mathbb{E} \left\| w_k - w^* \right\| \right)^2 + \frac{\tau}{\tilde{\mu} + \tau} \mathbb{E} \left\| w_k - w^* \right\|,$$

which gives us the linear-quadratic convergence. Moreover, if $\tau = 0$ and $\tilde{\mu} > 0$, we have the following quadratic convergence

$$\mathbb{E} \left\| w_{k+1} - w^* \right\| \leq \frac{\gamma (M + 2L + 2L^2)}{2\tilde{\mu}} \left(\mathbb{E} \left\| w_k - w^* \right\| \right)^2,$$

given that $\left\| w_0 - w^* \right\| \leq \frac{2\tilde{\mu}}{\gamma (M + 2L + 2L^2)}$. Here, convergence is guaranteed as

$$\mathbb{E} \left\| w_k - w^* \right\| \leq \frac{2\tilde{\mu}}{\gamma (M + 2L + 2L^2)} \quad \text{for all } k$$

by induction using the neighbourhood criterion.

□

A.3.1 Proof of Corollary 1

We restate Corollary 1.

Corollary 1 (Local quadratic convergence)**.** *Under the same assumptions as Theorem 2, if we decrease the regularization term according to* $\tau_k \leq \|\nabla f(w_k)\|$*,* R-SSN *can achieve local quadratic convergence with* $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau_k)}{\gamma(M+4L+2L^2)}$*:*

$$\mathbb{E}\,\|w_{k+1} - w^*\| \leq \frac{\gamma(M + 4L + 2L^2)}{2(\tilde{\mu}_k + \tau_k)}\,\left(\mathbb{E}\,\|w_k - w^*\|\right)^2,$$

where $\tilde{\mu}_k$ *is the minimum eigenvalue of* $\nabla^2 f_{\mathcal{S}_k}(w_k)$ *over all batches of size* $|\mathcal{S}_k|$*.*

Proof. By a similar analysis of Theorem 2 but replacing τ with τ_k and using $\tilde{\mu}_k$ instead of $\tilde{\mu}$, we arrive at

$$\begin{aligned}\mathbb{E}_k\,\|w_{k+1} - w^*\| \leq \frac{1}{(\tilde{\mu}_k + \tau_k)}\Big[&\underbrace{\left\|\nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k)\right\|}_{\text{(1) Bound using Lipschitz Hessian}} \\ &+ \underbrace{\mathbb{E}_k\left\|\left(\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right)(w_k - w^*)\right\|}_{\text{(2) Bound using Hessian Variance}} \\ &+ \underbrace{\mathbb{E}_k\,\|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\|}_{\text{(3) Bound using SGC}}\Big].\end{aligned}$$

The second term can then be bounded as

$$\begin{aligned}&\mathbb{E}_k\left\|\left(\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right)(w_k - w^*)\right\| \\ &\qquad\leq \|w_k - w^*\|\,\mathbb{E}_k\left\|\mathbf{H}_{\mathcal{S}_k}(w_k) - \nabla^2 f(w_k)\right\| \\ &\qquad\leq \|w_k - w^*\|\,\left[\|\nabla f(w_k)\| + \tau_k\right] \\ &\qquad\leq L\,\|w_k - w^*\|^2 + \tau_k\,\|w_k - w^*\|,\end{aligned}$$

and if we decrease the regularization factor as $\tau_k \leq \|\nabla f(w_k)\|$,

$$\begin{aligned}&\leq L\,\|w_k - w^*\|^2 + \|\nabla f(w_k)\|\,\|w_k - w^*\| \\ &\leq 2L\,\|w_k - w^*\|^2.\end{aligned}$$

The other two terms will be bounded similarly as in Theorem 2. Putting all three bounds together,

$$\begin{aligned}\mathbb{E}_k \left\| w_{k+1} - w^* \right\| &\leq \frac{1}{(\tilde{\mu}_k + \tau_k)} \Big[\frac{M}{2} \left\| w_k - w^* \right\|^2 \\ &\qquad + 2L \left\| w_k - w^* \right\|^2 + L^2 \left\| w_k - w^* \right\|^2 \Big] \\ &\leq \frac{\left(M + 4L + 2L^2\right)}{2(\tilde{\mu}_k + \tau_k)} \left\| w_k - w^* \right\|^2 .\end{aligned}$$

Using the γ bounded moments assumption,

$$\implies \mathbb{E} \left\| w_{k+1} - w^* \right\| \leq \frac{\gamma \left(M + 4L + 2L^2\right)}{2(\tilde{\mu}_k + \tau_k)} \left(\mathbb{E} \left\| w_k - w^* \right\| \right)^2 ,$$

which will converge in a local neighbourhood $\left\| w_k - w^* \right\| \leq \frac{2(\tilde{\mu} + \min_k \tau_k)}{\gamma(M + 4L + 2L^2)}$. □

A.3.2 Local quadratic convergence under the stronger SGC

Assumption 8 (Stronger SGC)**.** *A differentiable function f with a finite-sum structure satisfies the stronger-strong growth condition (*SGC*) if there exists $\rho \geq 1$ such that for all w,*

$$\mathbb{E}_i \left\| \nabla f_i(w) \right\|^2 \leq \rho \left\| \nabla f(w) \right\|^4 .$$

Corollary 2. *Suppose we replace the* SGC *assumption by Stronger* SGC *in Theorem 2 and let the rest be satisfied. Then the sequence $\{w_k\}_{k \geq 0}$ generated by* R-SSN *with (i) unit step-size $\eta_k = \eta = 1$ and (ii) constant batch-size $b_{g_k} = b_g$ for the gradients with a growing batch-size for Hessian such that*

$$b_{s_k} \geq \frac{n}{\frac{n}{\sigma^2} \left\| \nabla f(w_k) \right\| + 1}$$

converges to w^* with the quadratic rate

$$\mathbb{E}_k \left\| w_{k+1} - w^* \right\| \leq \gamma \left(\frac{M + 2L + 2L^2 \, c_g}{2(\tilde{\mu} + \tau)} \right) \left(\mathbb{E} \left\| w_k - w^* \right\| \right)^2$$

from a close enough initialization w_0 *such that* $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau)}{\gamma\,(M+2L+2L^2\,c_g)}$, *where* $c_g = \sqrt{\frac{\rho(n-b_g)}{b_g\,(n-1)}}$.

Proof. Using Lemma 2 to bound the third term in the analysis of Theorem 2, we obtain

$$\begin{aligned}\mathbb{E}_k \left\|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f(w_k)\right\| &\leq \sqrt{\frac{\rho\,(n-b_{g_k})}{(n-1)\,b_{g_k}}}\left\|\nabla f(w_k)\right\|^2 \\ &\leq L^2\sqrt{\frac{\rho\,(n-b_{g_k})}{(n-1)\,b_{g_k}}}\left\|w_k - w^*\right\|^2.\end{aligned}$$

The first two terms are bounded the same way, giving us

$$\begin{aligned}\mathbb{E}_k \left\|w_{k+1} - w^*\right\| \leq \frac{1}{(\tilde{\mu}+\tau)}\Big[\frac{M}{2}\left\|w_k - w^*\right\|^2 + L\left\|w_k - w^*\right\|^2 \\ + \tau\left\|w_k - w^*\right\| + L^2 c_g\,\left\|w_k - w^*\right\|^2\Big]\end{aligned}$$

where $c_g = \sqrt{\frac{\rho\,(n-b_g)}{(n-1)\,b_g}}$ and b_g can remain fixed for all the iterations. Using the γ-bounded moments assumption,

$$\begin{aligned}\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \frac{\gamma\left(M+2L+2L^2 c_g\right)}{2(\tilde{\mu}+\tau)}\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2 \\ + \frac{\tau}{\tilde{\mu}+\tau}\mathbb{E}\left\|w_k - w^*\right\|\end{aligned}$$

which gives us the linear-quadratic convergence. Furthermore if $\tau = 0$, $\tilde{\mu} > 0$ and $\|w_0 - w^*\| \leq \frac{2(\tilde{\mu}+\tau)}{\gamma\,(M+2L+2L^2\,c_g)}$ yields the quadratic rate

$$\mathbb{E}\left\|w_{k+1} - w^*\right\| \leq \left(\frac{\gamma\left(M+2L+2L^2\,c_g\right)}{2(\tilde{\mu}+\tau)}\right)\left(\mathbb{E}\left\|w_k - w^*\right\|\right)^2.$$

□

A.4 Proof of Theorem 3

We define the local norm of a direction h with respect to the local Hessian as

$$\|h\|_x = \left\langle \nabla^2 f(x)h\,,\, h\right\rangle^{1/2} = \left\| \left[\nabla^2 f(x)\right]^{1/2} h \right\|.$$

The following two theorems are standard results for self-concordant functions needed in our analysis. We refer the reader to Nesterov [2018] for the proofs.

Theorem 5 (Theorem 5.1.9 in Nesterov [2018])**.** *Let f be a standard self-concordant function, $x, y \in dom f$, and $\|y - x\|_x < 1$. Then*

$$f(y) \leq f(x) + \langle \nabla f(x)\,,\, y - x\rangle + \omega_* \left(\|y - x\|_x\right).$$

Here, $\omega_*(t) = -t - \log(1 - t)$.

Theorem 6 (Theorem 5.2.1 in Nesterov [2018])**.** *Let the unregularized, deterministic Newton decrement at w be defined as*

$$\lambda^0(w) := \left\langle \nabla f(w)\,,\, \left[\nabla^2 f(w)\right]^{-1} \nabla f(w)\right\rangle^{1/2}.$$

If f is standard self-concordant and $\lambda^0(w) < 1$, then

$$f(w) - f(w^*) \leq \omega_* \left(\lambda^0(w)\right).$$

We now restate Theorem 3 and proceed to its proof.

Theorem 3 (Two-phased analysis)**.** *Suppose f is self-concordant and satisfies L-smoothness, and that the subsampled Hessians have bounded eigenvalues in the range $[\tilde{\mu}+\tau, \tilde{L}+\tau]$ with $\tilde{\mu} \geq 0$. Suppose Newton decrement* SGC *holds with parameter $\rho_{nd} = \frac{\rho L}{\tilde{\mu}+\tau}$. Then if the sequence $\{w_k\}_{k\in[0,m]}$ generated by* R-SSN *in Eq.* (3.8) *stay in the bounded set with radius D with*

$$\eta \in \left(0, \frac{c}{\rho_{nd}(1+\tilde{L}D/(\tilde{\mu}+\tau))}\right] \quad \textit{where} \quad c = \sqrt{\frac{\tilde{\mu}+\tau}{L}}, \tag{3.9}$$

and constant batch sizes converges to w^* from an arbitrary initialization w_0 at a rate characterized by

$$\mathbb{E}\left[f(w_{k+1})\right] \leq f(w_k) - \eta\,\delta\,\omega\left(\lambda_k\right).$$

Here $\delta \in (0,1]$ and the univariate function ω is defined as $\omega(t) = t - \ln(1+t)$. Furthermore, in the local neighbourhood where $\lambda_m \leq 1/6$, the sequence $\{w_k\}_{k\geq m}$ converges to w^* at a Q-linear rate, given by

$$\mathbb{E}\left[f(w_T)\right] - f(w^*) \leq \left(1 - \frac{\eta\delta}{1.26}\right)^{T-m} (\mathbb{E}\left[f(w_m)\right] - f(w^*)). \tag{3.10}$$

Proof. We first analyze the norm of the update direction wrt the local Hessian,

$$\begin{aligned}
&\|w_{k+1} - w_k\|_{w_k} \\
&\quad = \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\| \left[\nabla^2 f(w_k)\right]^{1/2} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\| \\
&\quad = \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\| \left[\nabla^2 f(w_k)\right]^{1/2} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \nabla f_{\mathcal{G}_k}(w_k) \right\| \\
&\quad \leq \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\| \left[\nabla^2 f(w_k)\right]^{1/2} \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \right\| \left\| \left[\mathbf{H}_{\mathcal{S}_k}(w_k)\right]^{-1/2} \nabla f_{\mathcal{G}_k}(w_k) \right\| \\
&\quad \leq \frac{c\eta\tilde{\lambda}_k}{1+\eta\tilde{\lambda}_k} \sqrt{\frac{L}{\tilde{\mu}+\tau}},
\end{aligned}$$

where the last inequality follows from our definition of the regularized stochastic Newton decrement and regularized subsampled Hessian. Substituting in the choice

of c gives us,

$$\|w_{k+1} - w_k\|_{w_k} \leq \frac{\eta\tilde{\lambda}_k}{1+\eta\tilde{\lambda}_k} \leq 1. \tag{A.4}$$

This allows us to analyze the sub-optimality in terms of objective values using Theorem 5,

$$\begin{aligned} f(w_{k+1}) \leq f(w_k) - \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\rangle \\ + \omega_* \left(\|w_{k+1} - w_k\|_{w_k} \right). \end{aligned}$$

We know that ω_* is strictly increasing on the positive domain, using Eq. (A.4),

$$\begin{aligned} f(w_{k+1}) &\leq f(w_k) - \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\rangle \\ &\qquad + \omega_* \left(\frac{\eta\tilde{\lambda}_k}{1+\eta\tilde{\lambda}_k} \right) \\ &\leq f(w_k) - \frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\rangle \\ &\qquad + \omega_* \left(\omega' \left(\eta\tilde{\lambda}_k \right) \right) \end{aligned}$$

since $\omega'(t) = \frac{t}{1+t}$. Now take expectation on both sides with respect to $\mathcal{G}_k$ and $\mathcal{S}_k$ conditioned on w_k,

$$\begin{aligned} &\mathbb{E}_{\mathcal{G}_k,\mathcal{S}_k}\left[f(w_{k+1})\right] \\ &\quad \leq f(w_k) - \mathbb{E}_{\mathcal{G}_k,\mathcal{S}_k}\left[\frac{c\eta}{1+\eta\tilde{\lambda}_k} \left\langle \nabla f(w_k)\,,\, [\mathbf{H}_{\mathcal{S}_k}(w_k)]^{-1} \nabla f_{\mathcal{G}_k}(w_k) \right\rangle \right] \\ &\qquad + \mathbb{E}_{\mathcal{G}_k,\mathcal{S}_k}\left[\omega_* \left(\omega' \left(\eta\tilde{\lambda}_k \right) \right) \right]. \end{aligned} \tag{A.5}$$

To lower bound the middle term, observe that

$$\begin{aligned}
\tilde{\lambda}_k &\leq \frac{1}{\sqrt{\tilde{\mu}+\tau}} \left\|\nabla f_{\mathcal{G}_k}(w_k)\right\| \\
&= \frac{1}{\sqrt{\tilde{\mu}+\tau}} \left\|\nabla f_{\mathcal{G}_k}(w_k) - \nabla f_{\mathcal{G}}(w^*)\right\| && \text{(by interpolation assumption)} \\
&\leq \frac{\tilde{L}+\tau}{\sqrt{\tilde{\mu}+\tau}} \left\|w_k - w^*\right\| && \text{(by smoothness on the batch)} \\
&\leq \frac{(\tilde{L}+\tau)D}{\sqrt{\tilde{\mu}+\tau}} \\
&:= \lambda_{\max},
\end{aligned}$$

which combined with Eq. (A.5) gives

$$\mathbb{E}_{\mathcal{G}_k,\mathcal{S}_k}\left[f(w_{k+1})\right] \leq f(w_k) - \frac{c\eta\lambda_k^2}{1+\eta\lambda_{\max}} + \mathbb{E}_{\mathcal{G}_k,\mathcal{S}_k}\left[\omega_*\left(\omega'\left(\eta\tilde{\lambda}_k\right)\right)\right].$$

Using $\omega_*\left(\omega'\left(t\right)\right) = t\omega'\left(t\right) - \omega\left(t\right)$ for $t \geq 0$, the last term can be bounded as

$$\begin{aligned}
\mathbb{E}\left[\omega_*\left(\omega'\left(\eta\tilde{\lambda}_k\right)\right)\right] &= \mathbb{E}\left[\eta\tilde{\lambda}_k\omega'\left(\eta\tilde{\lambda}_k\right) - \omega\left(\eta\tilde{\lambda}_k\right)\right] \\
&\leq \frac{\eta^2\mathbb{E}\left[\tilde{\lambda}_k^2\right]}{1+\eta\tilde{\lambda}_{\min}} - \mathbb{E}\left[\omega\left(\eta\tilde{\lambda}_k\right)\right],
\end{aligned}$$

where $\tilde{\lambda}_{\min} = \min_{w_k,\mathcal{G}_k,\mathcal{S}_k}\tilde{\lambda}_k$. Applying the Newton decrement SGC gives us

$$\mathbb{E}\left[\omega_*\left(\omega'\left(\eta\tilde{\lambda}_k\right)\right)\right] \leq \frac{\eta^2\rho_{\text{nd}}\lambda_k^2}{1+\eta\tilde{\lambda}_{\min}} - \mathbb{E}\left[\omega\left(\eta\tilde{\lambda}_k\right)\right]$$

with $\rho_{\text{nd}} = \frac{\rho L}{\tilde{\mu}+\tau}$. Combining this with Eq. (A.5) gives us

$$\begin{aligned}
\mathbb{E}\left[f(w_{k+1})\right] &\leq f(w_k) - \frac{c\eta\lambda_k^2}{1+\eta\lambda_{\max}} + \frac{\eta^2\rho_{\text{nd}}\lambda_k^2}{1+\eta\tilde{\lambda}_{\min}} - \mathbb{E}\left[\omega\left(\eta\tilde{\lambda}_k\right)\right] \\
&= f(w_k) - \eta\lambda_k^2\underbrace{\left(\frac{c}{1+\eta\lambda_{\max}} - \frac{\eta\rho_{\text{nd}}}{1+\eta\tilde{\lambda}_{\min}}\right)}_{(*)} - \mathbb{E}\left[\omega\left(\eta\tilde{\lambda}_k\right)\right].
\end{aligned} \tag{A.6}$$

For η in the range $0 < \eta \leq \frac{c}{\rho_{\text{nd}}(1+\lambda_{\max})-c\lambda_{\min}} \quad (\leq 1)$, we have

$$\begin{aligned} &\eta\rho_{\text{nd}}(1+\lambda_{\max}) - \eta c\lambda_{\min} \leq c \\ \implies &\eta\rho_{\text{nd}}(1+\eta\lambda_{\max}) \leq \eta\rho_{\text{nd}}(1+\lambda_{\max}) \leq c(1+\eta\lambda_{\min}) \\ &\implies \frac{\eta\rho_{\text{nd}}}{1+\eta\lambda_{\min}} \leq \frac{c}{1+\eta\lambda_{\max}}. \end{aligned}$$

The $\eta \leq 1$ claim comes from substituting in our choice of c and definition of ρ_{nd}, which gives us

$$\begin{aligned} \eta &\leq \frac{c}{\rho_{\text{nd}}(1+\lambda_{\max}) - c\lambda_{\min}} \\ &= \frac{1}{\frac{\rho(1+\lambda_{\max})}{\left(\frac{\tilde{\mu}+\tau}{L}\right)^{3/2}} - \lambda_{\min}}, \end{aligned}$$

which is less than 1 since $\rho > 1$ and $\lambda_{\max} > \lambda_{\min}$ for τ chosen small enough. Thus we can choose $0 < \eta \leq \frac{c}{\rho_{\text{nd}}(1+\lambda_{\max})} \left(\leq \frac{c}{\rho_{\text{nd}}(1+\lambda_{\max})-c\lambda_{\min}}\right)$ and upper bound (*) in (A.6) by 0. Now we have the following expected decrease of the function value for one update,

$$\mathbb{E}\left[f(w_{k+1})\right] \leq f(w_k) - \mathbb{E}\left[\omega\left(\eta\tilde{\lambda}_k\right)\right]$$

by the convexity of ω and using Jensen's inequality,

$$\begin{aligned} &\leq f(w_k) - \omega\left(\eta\mathbb{E}\left[\tilde{\lambda}_k\right]\right) \\ &= f(w_k) - \omega\left(\eta\mathbb{E}\left\|\mathbf{H}_{\mathcal{S}_k}(w_k)^{-1/2}\nabla f_{\mathcal{G}_k}(w_k)\right\|\right) \end{aligned}$$

apply Jensen's inequality using the convexity of $\|\cdot\|_H$ for some $H \succeq 0$ and that ω is monotonically increasing on the positive domain and the fact that $\mathcal{G}_k$ and $\mathcal{S}_k$ are independent batches,

$$\leq f(w_k) - \omega\left(\eta\left\|\mathbb{E}\,\mathbf{H}_{\mathcal{S}_k}(w_k)^{-1/2}\mathbb{E}\,\nabla f_{\mathcal{G}_k}(w_k)\right\|\right).$$

Using the fact that the inverse square root function is operator convex (Löwner-Heinz Theorem) [Carlen, 2010, Tropp, 2015] on a positive spectrum, we can apply the operator Jensen inequality to bound the inner term, implying

$$\begin{aligned}\mathbb{E}[f(w_{k+1})] &\leq f(w_k) - \omega\left(\eta \left\| \left[\nabla^2 f(w_k) + \tau I_d\right]^{-1/2} \nabla f(w_k) \right\|\right) \\ &= f(w_k) - \omega\left(\eta\lambda_k\right). \end{aligned} \tag{A.7}$$

Note that for any $c, \delta \in (0, 1]$ and $t \geq 0$,

$$\begin{aligned}\omega\left(ct\right) - c\delta\omega\left(t\right) &= ct - \log(1+ct) - c\delta t + c\delta\log(1+t) \\ &\geq ct - c\log(1+t) - c\delta t + c\delta\log(1+t) \\ &= (c - c\delta)t - (c - c\delta)\log(1+t) \\ &= (c - c\delta)(t - \log(1+t)) \\ &= (c - c\delta)\omega\left(t\right) \\ &\geq 0 \\ \implies \omega\left(ct\right) &\geq c\delta\omega\left(t\right).\end{aligned}$$

Combining this with Eq. (A.7) yields the global R-linear convergence rate,

$$\begin{aligned}\mathbb{E}\left[f(w_{k+1})\right] &\leq f(w_k) - \eta\delta\omega\left(\lambda_k\right) \\ \implies \mathbb{E}\left[f(w_T)\right] - f(w^*) &\leq f(w_0) - f(w^*) - \eta\delta\left(\sum_{k=0}^{T-1}\omega\left(\lambda_k\right)\right).\end{aligned}$$

Note that $\lambda_k = \left\langle \nabla f(w_k), \left[\nabla^2 f(w_k) + \tau I\right]^{-1} \nabla f(w_k) \right\rangle^{1/2} \leq \lambda_k^0$, and since $\omega_*\left(t\right)$ is a decreasing function for $t \leq 1/6$, then $\omega_*\left(\lambda_k\right) \geq \omega_*\left(\lambda_k^0\right)$. As shown in Zhang and Lin [2015], for all $t \leq 1/6$, $\omega_*\left(t\right) \leq 1.26\,\omega\left(t\right)$, then for $\lambda_k \leq \lambda_k^0 \leq 1/6$, we can bound the above as

$$\mathbb{E}\left[f(w_{k+1})\right] \leq f(w_k) - \frac{\eta\delta}{1.26}\omega_*\left(\lambda_k^0\right).$$

Subtract $f(w^*)$ from both sides,

$$\leq f(w_k) - f(w^*) - \frac{\eta\delta}{1.26}\omega_* \left(\lambda_k^0\right)$$

and apply Theorem 6,

$$\leq f(w_k) - f(w^*) - \frac{\eta\delta}{1.26}(f(w_k) - f(w^*))$$

$$\mathbb{E}\left[f(w_{k+1})\right] - f(w^*) \leq \left(1 - \frac{\eta\delta}{1.26}\right)(f(w_k) - f(w^*))$$

which completes the proof. □

A.5 Proof of Theorem 4

We restate Theorem 4.

Theorem 4 (Global linear convergence of stochastic BFGS)**.** *Let μ-strong convexity, L-smoothness, and ρ-*SGC *be satisfied, and suppose the eigenvalues of $\mathbf{B}_k$ are bounded in $[\lambda_1, \lambda_d]$. Then the sequence $\{w_k\}_{k\geq 0}$ generated by stochastic* BFGS *with constant step-size $\eta_k = \eta = \frac{\lambda_1}{c_g L \lambda_d^2}$ and constant batch size $b_{g_k} = b_g$ converges to w^* at a linear rate from an arbitrary initialization w_0,*

$$\mathbb{E}[f(w_T)] - f(w^*) \leq \left(1 - \frac{\mu\lambda_1^2}{c_g\, L\lambda_d^2}\right)^T (f(w_0) - f(w^*))$$

where $c_g = \frac{(n-b_g)\,(\rho-1)}{(n-1)\,b_g} + 1$.

Proof. From the L-smoothness assumption, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] &\leq f(w_k) - \eta_k \left\langle \nabla f(w_k)\,,\, \mathbf{B}_k\, \mathbb{E}_{\mathcal{G}_k}[\nabla f_{\mathcal{G}_k}(w_k)] \right\rangle + \frac{L}{2}\eta_k^2\, \mathbb{E}_{\mathcal{G}_k} \left\| \mathbf{B}_k\, \nabla f_{\mathcal{G}_k}(w_k) \right\|^2 \\
&\leq f(w_k) - \eta_k \left\langle \nabla f(w_k)\,,\, \mathbf{B}_k \nabla f(w_k) \right\rangle + \frac{L\,\lambda_d^2\,\eta_k^2}{2}\, \mathbb{E}_{\mathcal{G}_k} \left\| \nabla f_{\mathcal{G}_k}(w_k) \right\|^2 .
\end{aligned}$$

Bounding the last term using Lemma 1,

$$\mathbb{E}_{\mathcal{G}_k} \left\| \nabla f_{\mathcal{G}_k}(w_k) \right\|^2 \leq \left(\frac{(n-b_g)\,(\rho-1)}{(n-1)\,b_g} + 1 \right) \left\| \nabla f(w_k) \right\|^2 .$$

Denoting $\left(\frac{(n-b_g)\,(\rho-1)}{(n-1)\,b_g} + 1\right)$ as ρ', the expected decrease becomes

$$\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] \leq f(w_k) - \eta_k\lambda_1 \left\| \nabla f(w_k) \right\|^2 + \frac{\rho' L\,\lambda_d^2\,\eta_k^2}{2} \left\| \nabla f(w_k) \right\|^2 .$$

Let $\eta_k = \eta = \frac{\lambda_1}{\rho' L \lambda_d^2}$,

$$\begin{aligned}
\implies \mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] &\leq f(w_k) - \frac{\lambda_1^2}{\rho' L \lambda_d^2} \|\nabla f(w_k)\|^2 + \frac{\rho' L \lambda_d^2}{2} \frac{\lambda_1^2}{\rho'^2 L^2 \lambda_d^4} \|\nabla f(w_k)\|^2 \\
&= f(w_k) - \left(\frac{\lambda_1^2}{\rho' L \lambda_d^2} - \frac{\lambda_1^2}{2\rho' L \lambda_d^2} \right) \|\nabla f(w_k)\|^2 \\
&= f(w_k) - \frac{\lambda_1^2}{2\rho' L \lambda_d^2} \|\nabla f(w_k)\|^2 .
\end{aligned}$$

Subtracting $f(w^*)$ from both sides and apply strong convexity,

$$\begin{aligned}
\mathbb{E}_{\mathcal{G}_k}[f(w_{k+1})] - f(w^*) &\leq f(w_k) - f(w^*) - \frac{\mu \lambda_1^2}{\rho' L \lambda_d^2} (f(w_k) - f(w^*)) \\
&= \left(1 - \frac{\mu \lambda_1^2}{\rho' L \lambda_d^2} \right) (f(w_k) - f(w^*)) \\
&\leq \left(1 - \frac{\mu \, \lambda_1^2}{\left(\frac{(n - b_g)(\rho - 1)}{(n-1) b_g} + 1 \right) L \, \lambda_d^2} \right) (f(w_k) - f(w^*)).
\end{aligned}$$

After applying recursion gives us the desired result,

$$\mathbb{E}[f(w_T)] - f(w^*) \leq \left(1 - \frac{\mu \, \lambda_1^2}{\left(\frac{(n - b_g)(\rho - 1)}{(n-1) b_g} + 1 \right) L \, \lambda_d^2} \right)^T (f(w_0) - f(w^*)).$$

□

A.6 Notes on convergence rates

Suppose a sequence $\{x_k\}_{k\geq 0}$ converges to x^*, for $q = 1$, define the limit of the ratio of successive errors as

$$p := \lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^q}.$$

The rate is referred to as Q-sublinear when $p = 1$, Q-linear when $p \in (0, 1)$, and Q-superlinear when $p = 0$, where Q stands for quotient. If $q = 2$ and $p < \infty$, it is referred to as Q-quadratic convergence. We say the rate is linear-quadratic if

$$\|x_{k+1} - x^*\| \leq p_1 \|x_k - x^*\| + p_2 \|x_k - x^*\|^2$$

for $p_1 \in (0, 1)$ and $p_2 < \infty$. See Fig. A.1 for an illustration.

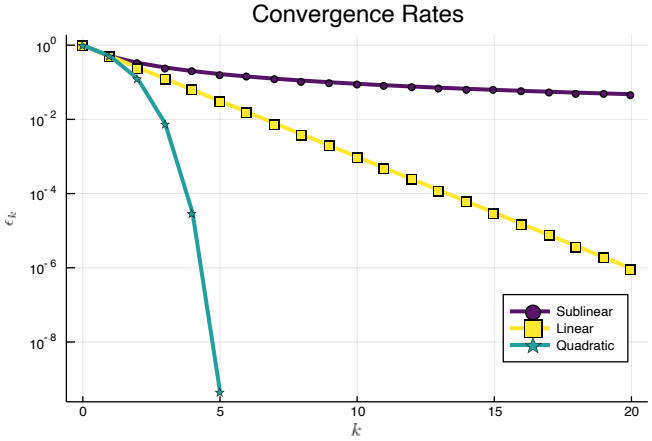

Figure A.1: Sequences constructed to depict different types of convergence rates in the quotient sense. For $k \geq 0$ and $c = 1/2$, the sublinear sequence is constructed as $\{1/(k + 1)\}$, and $\{c^k\}$ and $\{2 \cdot c^{2^k}\}$ for linear and superlinear, respectively.

Moreover, the R-linear convergence is a weaker notion, where R stands for root. It is characterized as the following: x_k is said to converge to x^* R-linearly if

there exists a sequence $\{\epsilon_k\}$ such that for all $k \geq 0$,

$$\|x_k - x^*\| \leq \epsilon_k$$

where $\{\epsilon_k\}$ converges Q-linearly to 0. Note that this is a less steady rate as it does not enforce a decrease at every step [Nocedal and Wright, 2006].