Do Subsampled Newton Methods Work for High-Dimensional Data?

Xiang Li,[1] **Shusen Wang,**[2] **Zhihua Zhang**[1,3]
[1]School of Mathematical Sciences, Peking University, China
[2]Department of Computer Science, Stevens Institute of Technology, USA
[3]National Engineering Lab for Big Data Analysis and Applications, Peking University, China
smslixiang@pku.edu.cn, shusen.wang@stevens.edu, zhzhang@math.pku.edu.cn

Abstract

Subsampled Newton methods approximate Hessian matrices through subsampling techniques to alleviate the per-iteration cost. Previous results require $\Omega(d)$ samples to approximate Hessians, where d is the dimension of data points, making it less practical for high-dimensional data. The situation is deteriorated when d is comparably as large as the number of data points n, which requires to take the whole dataset into account, making subsampling not useful. This paper theoretically justifies the effectiveness of subsampled Newton methods on strongly convex empirical risk minimization with high dimensional data. Specifically, we provably require only $\widetilde{\Theta}(d_{\text{eff}}^{\gamma})$ samples for approximating the Hessian matrices, where d_{eff}^{γ} is the γ-ridge leverage and can be much smaller than d as long as $n\gamma \gg 1$. Our theories work for three types of Newton methods: subsampled Netwon, distributed Newton, and proximal Newton.

Introduction

Let $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ be the feature vectors, $l_i(\cdot)$ is a convex, smooth, and twice differentiable loss function; the response y_i is captured by l_i. In this paper, we study the following optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} G(\mathbf{w}) := \frac{1}{n}\sum_{j=1}^{n} l_j(\mathbf{x}_j^T\mathbf{w}) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2 + r(\mathbf{w}) \qquad (1)$$

where $r(\cdot)$ is a non-smooth convex function. We first consider the simple case where $r(\cdot)$ is zero, i.e.,

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := \frac{1}{n}\sum_{j=1}^{n} l_j(\mathbf{x}_j^T\mathbf{w}) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2. \qquad (2)$$

Problem (2) arises frequently in machining learning (Shalev Shwartz and Ben David 2014). For example, in logistic regression, $l_j(\mathbf{x}_j^T\mathbf{w}) = \log(1+\exp(-y_j\mathbf{x}_j^T\mathbf{w}))$, and in linear regression, $l_j(\mathbf{x}_j^T\mathbf{w}) = \frac{1}{2}(\mathbf{x}_j^T\mathbf{w}-y_j)^2$. Then we consider the more general case where r is non-zero, e.g., LASSO (Tibshirani 1996) and elastic net (Zou and Hastie 2005).

To solve (2), many first order methods have been proposed. First-order methods solely exploit information in the objective function and its gradient. Accelerated gradient descent (Golub and Van Loan 2012; Nesterov 2013; Bubeck 2014), stochastic gradient descent (Robbins and Monro 1985), and their variants (Lin, Mairal, and Harchaoui 2015; Johnson and Zhang 2013; Schmidt, LeRoux, and Bach 2017) are the most popular approaches in practice due to their simplicity and low per-iteration time complexity. As pointed out by (Xu, Roosta Khorasan, and Mahoney 2017), the downsides of first-order methods are the slow convergence to high-precision and the sensitivity to condition number and hyper-parameters.

Second-order methods use not only the gradient but also information in the Hessian matrix in their update. In particular, the Newton's method, a canonical second-order method, has the following update rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t\mathbf{H}_t^{-1}\mathbf{g}_t, \qquad (3)$$

where the gradient $\mathbf{g}_t = \nabla F(\mathbf{w}_t)$ is the first derivative of the objective function at $\mathbf{w}_t$, the Hessian $\mathbf{H}_t = \nabla^2 F(\mathbf{w}_t)$ is the second derivative at $\mathbf{w}_t$, and α_t is the step size and can be safely set as one under certain conditions. In comparison to the first-order methods, Newton's method requires fewer iterations, is more robust to hyper-parameter settings, and is guaranteed super-linear local convergence to high-precision. However, Newton's method is slow in practice, as in each iteration many Hessian-vector products are required to solve the inverse problem $\mathbf{H}_t\mathbf{p} = \mathbf{g}_t$. Quasi-Newton methods use information from the history of updates to construct Hessian (Dennis and Moré 1977). Well-known works include Broyden-Fletcher-Goldfarb-Shanno (BFGS) (Wright and Nocedal 1999) and its limited memory version (L-BFGS) (Liu and Nocedal 1989), but their convergence rates are not comparable to Newton's method.

Recent works proposed the Sub-Sampled Newton (SSN) methods for reducing the per-iteration complexity of the Newton's method (Byrd et al. 2011; Pilanci and Wainwright 2015; Roosta-Khorasani and Mahoney 2016; Pilanci and Wainwright 2017; Xu, Roosta Khorasan, and Mahoney 2017; Berahas, Bollapragada, and Nocedal 2017; Ye, Luo, and Zhang 2017). For the particular problem (2), the Hessian matrix can be written in the form

$$\mathbf{H}_t = \frac{1}{n}\mathbf{A}_t^T\mathbf{A}_t + \gamma\mathbf{I}_d, \qquad (4)$$

for some $n \times d$ matrix $\mathbf{A}_t$ whose i-th row is a scaling of $\mathbf{x}_i$. The basic idea of SSN is to sample and scale s ($s \ll n$) rows of $\mathbf{A}$ to form $\widetilde{\mathbf{A}}_t \in \mathbb{R}^{s \times d}$ and approximate $\mathbf{H}_t$ by

$$\widetilde{\mathbf{H}}_t \; = \; \frac{1}{s} \widetilde{\mathbf{A}}_t^T \widetilde{\mathbf{A}}_t + \gamma \mathbf{I}_d,$$

The quality of Hessian approximation is guaranteed by random matrix theories (Tropp 2015; Woodruff 2014), based on which the convergence rate of SSN is established, e.g., (Pilanci and Wainwright 2017; Roosta-Khorasani and Mahoney 2016; Xu et al. 2016).

As the second-order methods perform heavy computation in each iteration and converge in a small number of iterations, they have been adapted to solve distributed machine learning aiming at reducing the communication cost (Shamir, Srebro, and Zhang 2014; Mahajan et al. 2015; Zhang and Lin 2015; Reddi et al. 2016; Wang et al. 2018). In particular, the Globally Improved Approximate NewTon Method (GIANT) (Wang et al. 2018) is based on the same idea as SSN and has fast convergence rate.

As well as Newton's method, SSN is not directly applicable to problem (1) because the objective function is non-smooth. Following the proximal-Newton method (Lee, Sun, and Saunders 2014), SSN has been adapted to solve convex optimization with non-smooth regularization (Liu et al. 2017). SSN has also been applied to optimize nonconvex problem (Xu, Roosta Khorasan, and Mahoney 2017; Tripuraneni et al. 2018).

Recall that n is the total number of samples, d is the number of features, and s is the size of the randomly sampled subset. (Suppose $s \ll n$; otherwise, the subsampling would not speed up computation.) The existing theories of SSN require s to be at least $\Omega(d)$. For the big-data setting, i.e., $d \ll n$, the existing theories nicely guarantee the convergence of SSN.

However, high-dimensional data is not uncommon at all in machine learning; d can be comparable to or even greater than n. Thus requiring both $s \ll n$ and $s = \Omega(d)$ seriously limits the application of SSN. We considers the question:

Do SSN and its variants work for (1) *when* $s < d$*?*

The empirical studies in (Xu et al. 2016; Xu, Roosta Khorasan, and Mahoney 2017; Wang et al. 2018) indicate that yes, SSN and its extensions have fast convergence even if s is substantially smaller than d. However, their empirical observations have not been verified by theory.

Our contributions

This work gives a definitive answer to the question (for a class of strongly convex optimization problems). We show it suffices to use $s = \tilde{\Theta}(d_{\text{eff}}^{\gamma})$ uniformly samples to approximate the Hessian, where γ is the regularization parameter, d_{eff}^{γ} ($\leq d$) is the γ-effective-dimension of the $d \times d$ Hessian matrix, and $\tilde{\Theta}$ hides the constant and logarithmic factors. If $n\gamma$ is larger than most of the d eigenvalues of the Hessian, then d_{eff}^{γ} is tremendously smaller than d (Cohen, Musco, and Musco 2015).

Our theory is applicable to three SSN methods: standard SSN, distributed Newton, and sub-sampled proximal Newton.

- We study the convex and smooth problem (2). we show the convergence of the standard SSN with the effective-dimension dependence and improves (Xu et al. 2016).
- For the problem (2), we extend the result to the distributed computing setting and improves the bound of GIANT (Wang et al. 2018).
- We study a convex but nonsmooth problem (1) and analyze the combination of SSN and proximal-Newton.

We additionally analyze SSN methods with the subproblems inexactly solved. The proofs of the main theorems are in the appendix.

Admittedly, this work has two limitations. First, this work is applicable to only strongly convex problems; we do not have theories for nonconvex problems.[1] Second, for non-quadratic objectives, we do not have global convergence bound. (We do have global convergence if the loss is quadratic.) They will be studied in our future work.

Notation and Preliminary

Basic matrix notation. Let $\mathbf{I}_n$ be the $n \times n$ indentity matrix. Let $\|\mathbf{a}\|_2$ denote the vector ℓ_2 norm and $\|\mathbf{A}\|_2$ denote the matrix spectral norm. Let

$$\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{5}$$

be its singular value decomposition (SVD), with $\sigma_{\max}(\mathbf{A})$ its largest singular value and $\sigma_{\min}(\mathbf{A})$ the smallest (the d-th largest). The moore-Penrose inverse of $\mathbf{A}$ is defined by $\mathbf{A}^{\dagger} = \mathbf{V} \boldsymbol{\Sigma}^{-1} \mathbf{U}^T$. If a symmetric real matrix has no negative eigenvalues, it is called symmetric positive semidefinite (SPSD). We denote $\mathbf{A} \preceq \mathbf{B}$ if $\mathbf{B} - \mathbf{A}$ is SPSD. For the SPSD matrice $\mathbf{H}$, we define a norm by $\|\mathbf{x}\|_{\mathbf{H}} = \sqrt{\mathbf{x}^T \mathbf{H} \mathbf{x}}$ and its conditional number by $\kappa(\mathbf{H}) = \frac{\sigma_{\max}(\mathbf{H})}{\sigma_{\min}(\mathbf{H})}$.

Ridge leverage scores. For $\mathbf{A} = [\mathbf{a}_1^T; \cdots; \mathbf{a}_n^T] \in \mathbb{R}^{n \times d}$, its row γ-ridge leverage score ($\gamma \geq 0$) is defined by

$$l_j^{\gamma} = \mathbf{a}_j^T (\mathbf{A}^T \mathbf{A} + n\gamma \mathbf{I}_d)^{\dagger} \mathbf{a}_j = \sum_{k=1}^{d} \frac{\sigma_k^2}{\sigma_k^2 + n\gamma} u_{jk}^2, \tag{6}$$

for $j \in [n] \triangleq \{1, 2, ..., n\}$. Here σ_k and $\mathbf{u}_k$ are defined in (5). For $\gamma = 0$, l_j^{γ} is the standard leverage score used by (Drineas, Mahoney, and Muthukrishnan 2008; Mahoney 2011).

Effective dimension. The γ-effective dimension of $\mathbf{A} \in \mathbb{R}^{n \times d}$ is defined by

$$d_{\text{eff}}^{\gamma}(\mathbf{A}) \; = \; \sum_{j=1}^{n} l_j^{\gamma} \; = \; \sum_{k=1}^{d} \frac{\sigma_k^2}{\sigma_k^2 + n\gamma} \; \leq \; d. \tag{7}$$

If $n\gamma$ is larger than most of the singular values of $\mathbf{A}^T \mathbf{A}$, then $d_{\text{eff}}^{\gamma}(\mathbf{A})$ is tremendously smaller than d (Alaoui and

[1] The Hessian matrix of a nonconvex problem is not SPSD. Thus effective dimension does not generalize to nonconvex problems straightforwardly.

Mahoney 2015; Cohen, Musco, and Musco 2017). In fact, to trade-off the bias and variance, the optimal setting of γ makes $n\gamma$ comparable to one of the top singular values of $\mathbf{A}^T\mathbf{A}$ (Hsu, Kakade, and Zhang 2014; Wang, Gittens, and Mahoney 2018), and thus $d_{\text{eff}}^{\gamma}(\mathbf{A})$ is small in practice.

Ridge coherence. The row γ-ridge coherence of $\mathbf{A} \in \mathbb{R}^{n\times d}$ is

$$\mu^{\gamma} \;=\; \frac{n}{d_{\text{eff}}^{\gamma}} \max_{i\in[n]} l_i^{\gamma}, \tag{8}$$

which measures the extent to which the information in the rows concentrates. If $\mathbf{A}$ has most of its mass in a relatively small number of rows, its γ-ridge coherence could be high. This concept is necessary for matrix approximation via uniform sampling. It could be imagined that if most information is in a few rows, which means high coherence, then uniform sampling is likely to miss some of the important rows, leading to low approximation quality. When $\gamma = 0$, it coincides with the standard row coherence

$$\mu^0 \;=\; \frac{n}{d} \max_{j\in[n]} l_j^0 \;=\; \frac{n}{d} \max_{j\in[n]} \mathbf{a}_j^T (\mathbf{A}^T\mathbf{A})^{\dagger} \mathbf{a}_j$$

which is widely used to analyze techniques such as compressed sensing (Candes, Romberg, and Tao 2006), matrix completion (Candès and Recht 2009), robust PCA (Candès et al. 2011), and so on.

Gradient and Hessian. For the optimization problem (2), the gradient of $F(\cdot)$ at $\mathbf{w}_t$ is

$$\mathbf{g}_t \;=\; \frac{1}{n}\sum_{j=1}^{n} l_j'(\mathbf{x}_j^T\mathbf{w}_t)\cdot \mathbf{x}_j \;+\; \gamma \mathbf{w}_t \;\in\; \mathbb{R}^d.$$

The Hessian matrix at $\mathbf{w}_t$ is

$$\mathbf{H}_t \;=\; \frac{1}{n}\sum_{j=1}^{n} l_j''(\mathbf{x}_j^T\mathbf{w}_t)\cdot \mathbf{x}_j\mathbf{x}_j^T \;+\; \gamma \mathbf{I}_d \;\in\; \mathbb{R}^{d\times d}.$$

Let $\mathbf{a}_j = \sqrt{l_j''(\mathbf{x}_i^T\mathbf{w}_t)}\cdot \mathbf{x}_j \in \mathbb{R}^d$ and

$$\mathbf{A}_t \;=\; [\mathbf{a}_1, \cdots, \mathbf{a}_n]^T \;\in\; \mathbb{R}^{n\times d}. \tag{9}$$

In this way, the Hessian matrix can be expressed as

$$\mathbf{H}_t \;=\; \tfrac{1}{n}\mathbf{A}_t^T\mathbf{A}_t + \gamma \mathbf{I}_d \;\in\; \mathbb{R}^{d\times d}. \tag{10}$$

Sub-Sampled Newton (SSN)

In this section, we provide new and stronger convergence guarantees for the SSN methods. For SSN with uniform sampling, we require a subsample size of $s = \tilde{\Theta}(\mu^{\gamma} d_{\text{eff}}^{\gamma})$. For SSN with ridge leverage score sampling,[2] a smaller sample size, $s = \tilde{\Theta}(d_{\text{eff}}^{\gamma})$, suffices. Because d_{eff}^{γ} is typically much smaller than d, our new results guarantee convergence when $s < d$.

Algorithm description

We set an interger s ($\ll n$) and uniformly sample s items out of $[n]$ to form the subset $\mathcal{S}$. In the t-th iteration, we form the matrix $\tilde{\mathbf{A}}_t \in \mathbb{R}^{s\times d}$ which contains the rows of $\mathbf{A}_t \in \mathbb{R}^{n\times d}$ indexed by $\mathcal{S}$ and the full gradient $\mathbf{g}_t$. Then, the approximately Newton direction $\tilde{\mathbf{p}}_t$ is computed by solving the linear system

$$\big(\tfrac{1}{s}\widetilde{\mathbf{A}}_t\widetilde{\mathbf{A}}_t^T + \gamma\mathbf{I}_d\big)\,\mathbf{p} \;=\; \mathbf{g}_t \tag{11}$$

by either matrix inversion or the conjugate gradient. Finally, $\mathbf{w}$ is updated by

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t - \alpha_t\tilde{\mathbf{p}}_t,$$

where α_t can be set to one or found by line search. In the rest of this section, we only consider $\alpha_t = 1$.

Most of the computation is performed in solving (11). The only difference between the standard Newton and the SSN methods is replacing $\mathbf{A}_t \in \mathbb{R}^{n\times d}$ by $\tilde{\mathbf{A}}_t \in \mathbb{R}^{s\times d}$. Measured by the per-iteration time complexity, SSN is $\frac{n}{s}$ times faster than Newton's method. However, SSN requires more iterations to converge. Nevertheless, to reach a fixed precision, the overall cost of SSN is much lower than Newton's method.

Our improved convergence bounds

Global convergence for quadratic loss. Let $\mathbf{w}^{\star}$ be the unique (due to the strong convexity) optimal solution to problem (1), $\mathbf{w}_t$ be the intermediate output of the t-th iteration, and $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^{\star}$. If the loss function of (1) is quadratic, e.g., $l_j(\mathbf{x}_j^T\mathbf{w}) = \frac{1}{2}(\mathbf{x}_j^T\mathbf{w} - y_j)^2$, the Hessian matrix $\mathbf{H}_t = \frac{1}{n}\mathbf{A}_t^T\mathbf{A}_t + \gamma\mathbf{I}_d$ does not change with the iteration, so we use $\mathbf{H}$ and $\mathbf{A}$ instead. Theorem 1 guarantees the global convergence of SSN.

Theorem 1 (Global Convergence)**.** *Let $d_{\textit{eff}}^{\gamma}$ and μ^{γ} respectively be the γ-ridge leverage score and γ-coherence of $\mathbf{A}$, and κ be the condition number of $\mathbf{H}$. Let $\varepsilon \in (0, \frac{1}{4})$ and $\delta \in (0,1)$ be any user-specified constants. Assume the loss function of* (1) *is quadratic. For a sufficiently large subsample size:*

$$s \;=\; \Theta\Big(\tfrac{\mu^{\gamma} d_{\textit{eff}}^{\gamma}}{\varepsilon^2} \log \tfrac{d_{\textit{eff}}^{\gamma}}{\delta}\Big),$$

with probability at least $1-\delta$,

$$\big\|\mathbf{\Delta}_t\big\|_2 \;\leq\; \epsilon^t\sqrt{\kappa}\,\big\|\mathbf{\Delta}_0\big\|_2. \tag{12}$$

Local convergence for non-quadratic loss. If the loss function of (1) is non-quadratic, the Hessian matrix $\mathbf{H}_t$ changes with iteration, and we can only guarantee fast local convergence, as well as the prior works (Roosta-Khorasani and Mahoney 2016; Xu et al. 2016). We make a standard assumption on the Hessian matrix, which is required by all the prior works on Newton-type methods.

Assumption 1. *The Hessian matrix $\nabla^2 F(\mathbf{w})$ is L-Lipschitz continuous, i.e., $\|\nabla^2 F(\mathbf{w}) - \nabla^2 F(\mathbf{w}')\|_2 \leq L\|\mathbf{w} - \mathbf{w}'\|_2$, for arbitrary $\mathbf{w}$ and $\mathbf{w}'$.*

Theorem 2 (Local Convergence)**.** *Let $d_{\textit{eff}}^{\gamma}$, μ^{γ} respectively be the γ-ridge leverage score and γ-coherence of $\mathbf{A}_t$. Let*

[2] We do not describe the ridge leverage score sampling in detail; the readers can refer to (Alaoui and Mahoney 2015; Cohen, Musco, and Musco 2015).

$\varepsilon \in (0, \frac{1}{4})$ and $\delta \in (0, 1)$ be any user-specified constants. Let Assumption 1 be satisfied. For a sufficiently large subsample size:

$$s \;=\; \Theta\Big(\tfrac{\mu^{\gamma} d_{\mathit{eff}}^{\gamma}}{\varepsilon^2} \log \tfrac{d_{\mathit{eff}}^{\gamma}}{\delta}\Big),$$

with probability at least $1 - \delta$,

$$\big\|\mathbf{\Delta}_{t+1}\big\|_2 \;\leq\; \varepsilon\sqrt{\kappa_t}\,\big\|\mathbf{\Delta}_t\big\|_2 + \tfrac{L}{\sigma_{\min}(\mathbf{H}_t)}\,\big\|\mathbf{\Delta}_t\big\|_2^2, \qquad (13)$$

where $\kappa_t = \frac{\sigma_{\max}(\mathbf{H}_t)}{\sigma_{\min}(\mathbf{H}_t)}$ is the condition number.

Theorem 3. *If ridge leverage score sampling is used instead, the sample complexity in Theorems 1 and 2 will be improved to*

$$s \;=\; \Theta\Big(\tfrac{d_{\mathit{eff}}^{\gamma}}{\varepsilon^2} \log \tfrac{d_{\mathit{eff}}^{\gamma}}{\delta}\Big).$$

Remark 1. *Ridge leverage score sampling eliminates the dependence on the coherence, and the bound is stronger than all the existing sample complexities for SSN. However, ridge leverage score sampling is expensive and impractical and thus has only theoretical interest.*

Although Newton-type methods empirically demonstrate fast global convergence (in terms of iterations) in almost all the real-world applications, they do not have stronger global convergence rate than first-order method. A weak global convergence bound for SSN was established by (Roosta-Khorasani and Mahoney 2016). We do not further discuss the global convergence issue in this paper.

Comparison with prior work

For SSN with uniform sampling, the prior work (Roosta-Khorasani and Mahoney 2016) showed that to obtain the same convergence bounds as ours, (12) and (13), the sample complexity should be

$$s \;=\; \Theta\Big(\tfrac{n\kappa_t}{\varepsilon^2(1-\epsilon\kappa_t)^2}\,\tfrac{\max_i \|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_2^2}\,\log\tfrac{d}{\delta}\Big).$$

In comparison, to obtain the same convergence rate, our sample complexity has a better dependence on the condition number and the dimensionality.

For the row norm square sampling of (Xu et al. 2016), which is slightly more expensive than uniform sampling, a sample complexity of

$$s \;=\; \tilde{\Theta}\Big(\tfrac{1}{\varepsilon^2(1-\epsilon\kappa_t)^2}\,\tfrac{\sigma_{\max}(\mathbf{A}_t^T\mathbf{A}_t)+n\gamma}{\sigma_{\max}(\mathbf{A}_t^T\mathbf{A}_t)}\,\sum_{i=1}^{d}\tfrac{\sigma_i(\mathbf{A}_t^T\mathbf{A}_t)}{\sigma_{\min}(\mathbf{A}_t^T\mathbf{A}_t)+n\gamma}\Big)$$

suffices for the same convergence rates as ours, (12) and (13). Their bound may or may not guarantee convergence for $s < d$. Even if $n\gamma$ is larger than most of the singular values of $\mathbf{A}_t^T\mathbf{A}_t$, their required sample complexity can be large.

For leverage score sampling, (Xu et al. 2016) showed that to obtain the same convergence bounds as ours, (12) and (13), the sample complexity should be

$$s \;=\; \Theta\big(\tfrac{d}{\varepsilon^2}\,\log\tfrac{d}{\delta}\big),$$

which depends on d (worse than ours d_{eff}^{γ}) but does not depend on coherence. We show that if the *ridge leverage score sampling* is used, then $s = \Theta\big(\frac{d_{\text{eff}}^{\gamma}}{\varepsilon^2}\log\frac{d_{\text{eff}}^{\gamma}}{\delta}\big)$ samples suffices, which is better than the above sample complexity. However, because approximately computing the (ridge) leverage scores is expensive, neither the leverage score sampling used by (Xu et al. 2016) nor the ridge leverage score sampling used by us is a practical choice.

SSN was studied by very recent work, e.g., (Kasai and Mishra 2018; Roosta et al. 2018; Tripuraneni et al. 2018; Zhou, Xu, and Gu 2018). They are less relevant to this work, so we do not discuss them in detail.

Distributed Newton-Type Method

Communication-efficient distributed optimization is an important research field, and second-order methods have been developed to reduce the communication cost, e.g., DANE (Shamir, Srebro, and Zhang 2014), AIDE (Reddi et al. 2016), DiSCO (Zhang and Lin 2015), and GIANT (Wang et al. 2018). Among them, GIANT has the strongest convergence bound for strongly convex and smooth problems. In this section, we further improve the convergence analysis of GIANT and show that GIANT does converge when the local sample size, $s = \frac{n}{m}$, is smaller than the number of features, d.

Motivation and algorithm description

Assume n samples are partition among m worker machines *uniformly at random*. Each worker machine has its own processors and memory, and the worker machines can communicate by message passing. The communication are costly compared to local computation; when the number of worker machines is large, communication is oftentimes the bottleneck of distributed computing. Thus there is a strong desire to reduce the communication cost of distributed computing. Our goal is to solve the optimization problem (2) in a communication-efficient way.

The first-order methods are computation-efficient but not communication-efficient. Let us take gradient descent for example. In each iteration, with the iteration $\mathbf{w}_t$ at hand, the i-th worker machine uses its local data to compute a *local gradient* $\mathbf{g}_{t,i}$; then the driver machine averages the local gradient to form the exact gradient $\mathbf{g}_t$ and update the model by

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t - \alpha_t \mathbf{g}_t,$$

where α_t is the step size. Although each iteration is computationally efficient, the first-order methods (even with acceleration) take many iterations to converge, especially when the condition number is big. As each iteration requires broadcasting $\mathbf{w}_t$ and aggregating the local gradients to form $\mathbf{g}_t = \sum_{i=1}^{m} \mathbf{g}_{t,i}$, the total number and complexity of communication are big.

Many second-order methods have been developed to improve the communication-efficiency, among which the Globally Improved Approximate NewTon (GIANT) method (Wang et al. 2018) has the strongest convergence rates, provided that the objective is strongly convex and smooth. Let $s = \frac{n}{m}$ be the local sample size and $\mathbf{A}_{t,i} \in \mathbb{R}^{s\times d}$ be the i-th block of $\mathbf{A}_t \in \mathbb{R}^{n\times d}$, which is previously defined in (9). With the iteration $\mathbf{w}_t$ at hand, the i-th worker machine can use its local data samples to form the *local Hessian matrix*

$$\widetilde{\mathbf{H}}_{t,i} \;=\; \tfrac{1}{s}\mathbf{A}_{t,i}^T\mathbf{A}_{t,i} + \gamma\mathbf{I}_d$$

and outputs the local *Approximate NewTon (ANT)* direction

$$\widetilde{\mathbf{p}}_{t,i} \;=\; \widetilde{\mathbf{H}}_{t,i}^{-1}\mathbf{g}_t. \tag{14}$$

Finally, the driver machine averages the ANT directions to get

$$\widetilde{\mathbf{p}}_t \;=\; \frac{1}{m}\sum_{i=1}^{m}\widetilde{\mathbf{p}}_{t,i}$$

and perform the update

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t - \alpha_t\widetilde{\mathbf{p}}_t,$$

where the step size α_t can be set to one under certain conditions; we only consider the $\alpha_t = 1$ case in this paper.

GIANT is much more communication-efficient than the first-order methods. With α_t fixed, each iteration of GIANT has four rounds of communications: (1) broadcasting $\mathbf{w}_t$, (2) aggregating the local gradients to form $\mathbf{g}_t$, (3) broadcasting $\mathbf{g}_t$, and (4) aggregating the ANT directions to form $\widetilde{\mathbf{p}}_t$. Thus the per-iteration communication cost is just twice as much as a first-order method. Wang et al. showed that GIANT requires a much smaller number of iterations than the accelerated gradient method which has the optimal iteration complexity (without using second-order information).

Our improved convergence bounds

We analyze GIANT and improve the convergence analysis of (Wang et al. 2018). Throughout this section, we assume the n samples are partitioned to m worker machine uniformly at random.

Global convergence for quadratic loss. We let $\mathbf{w}^\star$ be the unique optimal solution to problem (2) and $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^\star$. If the loss function of (2) is quadratic, e.g., $l_i(\mathbf{x}_i^T\mathbf{w}) = \frac{1}{2}(\mathbf{x}_i^T\mathbf{w} - y_i)^2$, the Hessian matrix $\mathbf{H}_t = \frac{1}{n}\mathbf{A}_t^T\mathbf{A}_t + \gamma\mathbf{I}_d$ does not change with the iteration, so we use $\mathbf{H}$ and $\mathbf{A}$ instead. Theorem 4 guarantees the global convergence of GIANT.

Theorem 4 (Global Convergence)**.** *Let $d_{\textit{eff}}^\gamma, \mu^\gamma$ respectively be the γ-ridge leverage score and γ-coherence of $\mathbf{A}$, and κ be the condition number of $\mathbf{H}$. Let $\varepsilon \in (0, \frac{1}{4})$ and $\delta \in (0, 1)$ be any user-specified constants. Assume the loss function of* (1) *is quadratic. For a sufficiently large sub-sample size:*

$$s \;=\; \Theta\Big(\tfrac{\mu^\gamma d_{\textit{eff}}^\gamma}{\varepsilon}\log\tfrac{md_{\textit{eff}}^\gamma}{\delta}\Big),$$

with probability at least $1-\delta$,

$$\big\|\mathbf{\Delta}_t\big\|_2 \;\le\; \varepsilon^t\sqrt{\kappa}\,\big\|\mathbf{\Delta}_0\big\|_2. \tag{15}$$

Local convergence for non-quadratic loss. If the loss function of (1) is non-quadratic, we can only guarantee fast local convergence under Assumption 1, as well as the prior works (Wang et al. 2018).

Theorem 5 (Local Convergence)**.** *Let $d_{\textit{eff}}^\gamma, \mu^\gamma$ respectively be the γ-ridge leverage score and γ-coherence of $\mathbf{A}_t$. Let $\varepsilon \in (0, \frac{1}{4})$ and $\delta \in (0, 1)$ be any user-specified constants. Let Assumption 1 be satisfied. For a sufficiently large sub-sample size:*

$$s \;=\; \Theta\Big(\tfrac{\mu^\gamma d_{\textit{eff}}^\gamma}{\varepsilon}\log\tfrac{md_{\textit{eff}}^\gamma}{\delta}\Big),$$

with probability at least $1-\delta$,

$$\big\|\mathbf{\Delta}_{t+1}\big\|_2 \;\le\; \varepsilon\sqrt{\kappa_t}\,\big\|\mathbf{\Delta}_t\big\|_2 + \tfrac{L}{\sigma_{\min}(\mathbf{H}_t)}\,\big\|\mathbf{\Delta}_t\big\|_2^2, \tag{16}$$

where $\kappa_t = \frac{\sigma_{\max}(\mathbf{H}_t)}{\sigma_{\min}(\mathbf{H}_t)}$ is the condition number.

Remark 2. *GIANT is a variant of SSN: SSN uses one of $\{\widetilde{\mathbf{p}}_{t,i}\}_{i=1}^m$ as the descending direction, whereas GIANT uses the averages of the m directions. As a benefit of the averaging, the sample complexity is improved from $s = \tilde{\Theta}\big(\frac{\mu^\gamma d_{\textit{eff}}^\gamma}{\epsilon^2}\big)$ to $s = \tilde{\Theta}\big(\frac{\mu^\gamma d_{\textit{eff}}^\gamma}{\epsilon}\big)$.*

Comparison with prior work

To guarantee the same convergence bounds, (15) and (16), Wang et al. require a sample complexity of $s = \Theta(\frac{\mu^0 d}{\varepsilon}\log\frac{d}{\delta})$.[3] This requires require the local sample size $s = \frac{n}{m}$ be greater than d, even if the coherence μ^0 is small. As communication and synchronization costs grow with m, the communication-efficient method, GIANT, is most useful for the large m setting; in this case, the requirement $n > md$ is unlikely satisfied.

In contrast, our improved bounds do not require $n > md$. As d_{eff}^γ can be tremendously smaller than d, our requirement can be satisfied even if m and d are both large. Our bounds match the empirical observation of (Wang et al. 2018): GIANT convergences rapidly even if md is larger than n.

The very recent work (Yuan and Li 2019) improves the convergence rate of DANE (Shamir, Srebro, and Zhang 2014). In particular, they showed global convergence rate for non-quadratic problem which matches accelerated gradient descent. However, for quadratic loss, their global convergence rate is worse than this work.

Sub-Sampled Proximal Newton (SSPN)

In previous sections, we analyze second-order methods for the optimization problem (2) which has a smooth objective function. In this section, we study the following problem which can be non-smooth:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{j=1}^{n} l_j(\mathbf{x}_j^T\mathbf{w}) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2 + r(\mathbf{w}),$$

where r is any convex function. The standard Newton's method does not apply because the second derivative of the objective function does not exist. Proximal Newton (Lee, Sun, and Saunders 2014), a second-order method, was developed to solve the problem, and later on, sub-sampling was incorporated to speed up computation (Liu et al. 2017). We further improve the bounds of Sub-Sampled Proximal Newton (SSPN).

Algorithm Description

Let $F(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} l_j(\mathbf{x}_j^T\mathbf{w}) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2$ be the smooth part of the objective function, and $\mathbf{g}_t$ and $\mathbf{H}_t$ be its first and second derivatives at $\mathbf{w}_t$. The proximal Newton method (Lee,

[3] The sample complexity in (Wang et al. 2018) is actually slightly worse; but it is almost trivial to improve their result to what we show here.

Sun, and Saunders 2014) iteratively solves the problem:

$$\mathbf{p}_t \; = \; \operatorname*{argmin}_{\mathbf{p}} \tfrac{1}{2}\big(\mathbf{p}^T\mathbf{H}_t\mathbf{p} - 2\mathbf{g}_t^T\mathbf{p} + \mathbf{g}_t^T\mathbf{H}_t^{-1}\mathbf{g}_t\big) + r(\mathbf{w}_t - \mathbf{p}),$$

and then perform the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \mathbf{p}_t$. The righthand side of the problem is a local quadratic approximation to $F(\mathbf{w})$ at $\mathbf{w}_t$. If $r(\cdot) = 0$, then proximal Newton will be the same as the standard Newton's method.

The sub-sampled proximal Newton (SSPN) method uses sub-sampling to approximate $\mathbf{H}_t$; let the approximate Hessian matrix be $\widetilde{\mathbf{H}}_t$, as previously defined in (4). SSPN computes the ascending direction by solving the local quadratic approximation

$$\widetilde{\mathbf{p}}_t \; = \; \operatorname*{argmin}_{\mathbf{p}} \tfrac{1}{2}\big(\mathbf{p}^T\widetilde{\mathbf{H}}_t\mathbf{p} - 2\mathbf{g}_t^T\mathbf{p} + \mathbf{g}_t^T\widetilde{\mathbf{H}}_t^{-1}\mathbf{g}_t\big) + r(\mathbf{w}_t - \mathbf{p}), \tag{17}$$

and then performs the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \widetilde{\mathbf{p}}_t$.

Our improved convergence bounds

We show that SSPN has exactly the same iteration complexity as SSN, for either quadratic or non-quadratic function $l_j(\cdot)$. Nevertheless, the overall time complexity of SSPN is higher than SSN, as the subproblem (17) is expensive to solve if $r(\cdot)$ is non-smooth.

Theorem 6. *Theorems 1, 2, and 3 hold for SSPN.*

Comparison with prior work

Liu et al. showed that when $\|\mathbf{\Delta}_t\|_2$ is small enough, $\|\mathbf{\Delta}_{t+1}\|_2$ will converge to zero linear-quadratically, similar to our results. But their sample complexity is

$$s = \tilde{\Theta}\big(\tfrac{d}{\varepsilon^2}\big).$$

This requires the sample size s greater than d. The ℓ_1 regularization, $r(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$, is often used for high-dimensional data, the requirement that $d < s \ll n$ violates the motivation of ℓ_1 regularization.

Our improved bounds show that $s = \tilde{\Theta}\big(\frac{d_{\text{eff}}^{\gamma}\mu^{\gamma}}{\varepsilon^2}\big)$ suffices for uniform sampling and that $s = \tilde{\Theta}\big(\frac{d_{\text{eff}}^{\gamma}}{\varepsilon^2}\big)$ suffices for ridge leverage score sampling. Since d_{eff}^{γ} can be tremendously smaller than d when $n\gamma \gg 1$, our bounds are useful for high-dimensional data.

Inexactly Solving the Subproblems

Each iteration of SSNand GIANT involves solving a subproblem in the form

$$\big(\tfrac{1}{s}\widetilde{\mathbf{A}}_t^T\widetilde{\mathbf{A}}_t + \gamma\mathbf{I}_d\big)\mathbf{p} \; = \; \mathbf{g}_t.$$

Exactly solving this problem would perform the multiplication $\widetilde{\mathbf{A}}_t^T\widetilde{\mathbf{A}}_t$ and inversion of the $d \times d$ approximate Hessian matrix $\frac{1}{s}\widetilde{\mathbf{A}}_t^T\widetilde{\mathbf{A}}_t + \gamma\mathbf{I}_d$; the time complexity is $\mathcal{O}(sd^2 + d^3)$.

In practice, the linear system can be approximately solved by the conjugate gradient (CG) method, each iteration of which applies a vector to $\widetilde{\mathbf{A}}_t$ and $\widetilde{\mathbf{A}}_t^T$; the time complexity is $\mathcal{O}(q \cdot \mathsf{nnz}(\mathbf{A}))$, where q is the number of CG iterations and nnz is the number of nonzeros. The inexact solution is particularly appealing if the data are sparse. In the following, we analyze the effect of the inexact solution of the subproblem.

Let κ_t be the condition number of $\widetilde{\mathbf{H}}_t$. Standard convergence bound of CG guarantees that by performing

$$q \;\approx\; \tfrac{\sqrt{\kappa_t}-1}{2}\log\tfrac{8}{\varepsilon_0^2}$$

CG iterations, the conditions (18) and (19) are satisfied, and the inexact solution does not much affect the convergence of SSN and GIANT.

Corollary 7 (SSN)**.** *Let $\widetilde{\mathbf{p}}_t$ and $\widetilde{\mathbf{p}}_t'$ be respectively the exact and an inexact solution to the quadratic problem $\widetilde{\mathbf{H}}_t^{-1}\mathbf{p} = \mathbf{g}_t$. SSN updates $\mathbf{w}$ by $\mathbf{w}_{t+1} = \mathbf{w}_t - \tilde{\mathbf{p}}_t'$. If the condition*

$$\big\|\widetilde{\mathbf{H}}_t^{1/2}\,(\widetilde{\mathbf{p}}_t - \widetilde{\mathbf{p}}_t')\big\|_2 \;\leq\; \tfrac{\varepsilon_0}{2}\big\|\widetilde{\mathbf{H}}_t^{1/2}\,\widetilde{\mathbf{p}}_t\big\|_2 \tag{18}$$

is satisfied for some $\varepsilon_0 \in (0,1)$, then Theorems 1 and 2, with ε in (12) *and* (13) *replaced by $\varepsilon + \varepsilon_0$, will continue holding.*

Corollary 8 (GIANT)**.** *Let $\tilde{\mathbf{p}}_{t,i}$ and $\tilde{\mathbf{p}}_{t,i}'$ be respectively the exact and an inexact solution to the quadratic problem $\widetilde{\mathbf{H}}_{t,i}^{-1}\mathbf{p} = \mathbf{g}_t$. GIANT updates $\mathbf{w}$ by $\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{1}{m}\sum_{i=1}^{m}\widetilde{\mathbf{p}}_{t,i}'$. If the condition*

$$\big\|\widetilde{\mathbf{H}}_{t,i}^{1/2}\,(\widetilde{\mathbf{p}}_{t,i} - \widetilde{\mathbf{p}}_{t,i}')\big\|_2 \;\leq\; \tfrac{\varepsilon_0}{2}\big\|\widetilde{\mathbf{H}}_t^{1/2}\,\widetilde{\mathbf{p}}_{t,i}\big\|_2 \tag{19}$$

is satisfied for some $\varepsilon_0 \in (0,1)$ and all $i \in [m]$, then Theorems 4 and 5, with ε in (15) *and* (16) *replaced by $\varepsilon + \varepsilon_0$, will continue holding.*

SSPN is designed for problems with non-smooth regularization, in which case finding the exact solution may be infeasible, and the sub-problem can only be inexactly solved. If the inexact solution satisfies the same condition as (18), Corollary 9 will guarantee the convergence of SSPN.

Corollary 9 (SSPN)**.** *Let $\tilde{\mathbf{p}}_t$ and $\tilde{\mathbf{p}}_t'$ be respectively the exact and an inexact solution to the non-smooth problem* (17)*. SSPN updates $\mathbf{w}$ by $\mathbf{w}_{t+1} = \mathbf{w}_t - \tilde{\mathbf{p}}_t'$. If $\widetilde{\mathbf{p}}_t'$ satisfies the condition* (18) *for any $\varepsilon_0 \in (0,1)$, then Theorem 6 still holds for SSPN with ε replaced by $\varepsilon + \varepsilon_0$.*

Conclusion

We studied the subsampled Newton (SSN) method and its variants, GIANT and SSPN, and established stronger convergence guarantees than the prior works. In particular, we showed that a sample size of $s = \tilde{\Theta}(d_{\text{eff}}^{\gamma})$ suffices, where γ is the ℓ_2 regularization parameter and d_{eff}^{γ} is the effective dimension. When $n\gamma$ is larger than most of the eigenvalues of the Hessian matrices, d_{eff}^{γ} is much smaller than the dimension of data, d. Therefore, our work guarantees the convergence of SSN, GIANT, and SSPN on high-dimensional data where d is comparable to or even greater than n. In contrast, all the prior works required a conservative sample size $s = \Omega(d)$ to attain the same convergence rate as ours. Because subsampling means that s is much smaller than n, the prior works did not lend any guarantee to SSN on high-dimensional data.

Admittedly, our theories are limited to strongly convex problems. We established global convergence for quadratic loss and local convergence for non-quadratic loss. However, we do not have global convergence guarantee for non-quadratic loss. Nonconvex problems and global convergence will be left as our future work.

Sketch of Proof

Random Sampling for Matrix Approximation

Here, we give a short introduction to random sampling and their theoretical properties. Given a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, row selection constructs a smaller matrix $\mathbf{C} \in \mathbb{R}^{s \times d}$ $(s < n)$ as an approximation of $\mathbf{A}$. The rows of $\mathbf{C}$ is constructed using a randomly sampled and carefully scaled subset of the rows of $\mathbf{A}$. Let $p_1, \cdots, p_n \in (0,1)$ be the sampling probabilities associated with the rows of $\mathbf{A}$. The rows of $\mathbf{C}$ are selected independently according to the sampling probabilities $\{p_j\}_{j=1}^n$:

$$\mathbb{P}(\mathbf{c}_j = \mathbf{a}_k / \sqrt{sp_k}) = p_k,$$

where $\mathbf{c}_j$ and $\mathbf{a}_k$ are the j-th row of $\mathbf{C}$ and k-th row of $\mathbf{A}$. In a matrix multiplication form, $\mathbf{C}$ can be formed as

$$\mathbf{C} = \mathbf{S}^T \mathbf{A},$$

where $\mathbf{S} \in \mathbb{R}^{\mathbf{s} \times d}$ is called the sketching matrix. As a result of row selection, there is only one non-zero entry in each column of $\mathbf{S}$, whose position and value correspond to the sampled row of $\mathbf{A}$.

Ridge leverage score sampling. It takes p_j proportional to the j-th ridge leverage score:

$$p_j = l_j^\gamma / \sum_{i=1}^n l_i^\gamma, \quad \forall\, j \in [n] \tag{20}$$

where l_i^γ is the ridge leverage score of the i-th row of $\mathbf{A}$. Let $\mathbf{U}$ be its sketching matrix. Then the non-zero entry in j-th column of $\mathbf{U}$ is $\sqrt{\frac{1}{s \cdot p_k}}$ if the j-th row of $\mathbf{U}^T\mathbf{A}$ is drawn from the k-th row of $\mathbf{A}$, where p_k is defined as (20). If ridge leverage score sampling is used to approximate the $d \times d$ Hessian matrix, the approximate Hessian matrix turns to

$$\widetilde{\mathbf{H}}_t \;=\; \tfrac{1}{n}\mathbf{A}_t^T\mathbf{U}\mathbf{U}^T\mathbf{A}_t + \gamma\mathbf{I}_d. \tag{21}$$

The sample complexity in Theorems 1 and 2 will be improved to $s = \Theta\big(\frac{d_{\text{eff}}^\gamma}{\varepsilon^2}\log\frac{d_{\text{eff}}^\gamma}{\delta}\big)$. Lemma 10 and 11 can be similarly proved by following (Cohen, Musco, and Musco 2015)

Lemma 10 (Ridge Leverage Rampling)**.** *Let* $\mathbf{H}_t$ *and* $\widetilde{\mathbf{H}}_t$ *be defined in* (10) *and* (21)*. Denote* $d_{\mathit{eff}}^\gamma = d_{\mathit{eff}}^\gamma(\mathbf{A}_t), \mu^\gamma = \mu^\gamma(\mathbf{A}_t)$ *for simplicity. Given arbitrary error tolerance* $\varepsilon \in (0,1)$ *and failure probability* $\delta \in (0,1)$*, for*

$$s = \Theta\left(\frac{d_{\mathit{eff}}^\gamma}{\varepsilon^2}\log\frac{d_{\mathit{eff}}^\gamma}{\delta}\right),$$

the spectral approximation holds with probability at least $1-\delta$*:*

$$(1-\varepsilon)\mathbf{H}_t \preceq \widetilde{\mathbf{H}}_t \preceq (1+\varepsilon)\mathbf{H}_t.$$

Uniform sampling. Uniform sampling simply sets all the sampling probabilities equal, i.e., $p_1 = \cdots = p_n = \frac{1}{n}$. Its corresponding sketching matrix $\mathbf{S}$ is often called uniform sampling matrix. The non-zero entry in each column of $\mathbf{S}$ is the same, i.e., $\sqrt{\frac{n}{s}}$. If s is sufficiently large,

$$\widetilde{\mathbf{H}}_t \;=\; \tfrac{1}{n}\mathbf{A}_t^T\mathbf{S}\mathbf{S}^T\mathbf{A}_t + \gamma\mathbf{I}_d \tag{22}$$

is a good approximation to $\mathbf{H}_t$.

Lemma 11 (Uniform Sampling)**.** *Let* $\mathbf{H}_t$ *and* $\widetilde{\mathbf{H}}_t$ *be defined as that in* (10) *and* (22)*. Denote* $d_{\mathit{eff}}^\gamma = d_{\mathit{eff}}^\gamma(\mathbf{A}_t), \mu^\gamma = \mu^\gamma(\mathbf{A}_t)$ *for simplicity. Given arbitrary error tolerance* $\varepsilon \in (0,1)$ *and failure probability* $\delta \in (0,1)$*, for*

$$s = \Theta\left(\frac{\mu^\gamma d_{\mathit{eff}}^\gamma}{\varepsilon^2}\log\frac{d_{\mathit{eff}}^\gamma}{\delta}\right)$$

the spectral approximation holds with probability at least $1-\delta$*:*

$$(1-\varepsilon)\mathbf{H}_t \preceq \widetilde{\mathbf{H}}_t \preceq (1+\varepsilon)\mathbf{H}_t.$$

Analyzing SSN, GIANT, and SSPN

To show the convergence of SSN and GIANT, we define the quadratic auxiliary function

$$\phi_t(\mathbf{p}) \triangleq \mathbf{p}^T \underbrace{(\frac{1}{n}\mathbf{A}_t^T\mathbf{A}_t + \gamma\mathbf{I}_d)}_{\triangleq \mathbf{H}_t} \mathbf{p} - 2\mathbf{p}^T\mathbf{g}_t.$$

If $\tilde{\mathbf{H}}_t$ well approximates $\mathbf{H}_t$, then measured by ϕ_t, the approximate Newton direction $\tilde{\mathbf{p}}$ is close to the exact Newton direction $\mathbf{p}$. Then, the convergence of SSN and GIANT follows from that $\phi_t(\tilde{\mathbf{p}}) \approx \phi_t(\mathbf{p})$. The convergence analysis of SSPN is similar but more involved.

Acknowledgments

Xiang Li and Zhihua Zhang have been supported by the National Natural Science Foundation of China (No. 11771002 and 61572017), Beijing Natural Science Foundation (Z190001), and Beijing Academy of Artificial Intelligence (BAAI).

References

Alaoui, A., and Mahoney, M. W. 2015. Fast Randomized Kernel Ridge Regression with Statistical Guarantees. In *Advances in Neural Information Processing Systems (NIPS)*.

Berahas, A. S.; Bollapragada, R.; and Nocedal, J. 2017. An investigation of Newton-sketch and subsampled Newton methods. *arXiv preprint arXiv:1705.06211*.

Bubeck, S. 2014. Theory of convex optimization for machine learning. *arXiv preprint arXiv:1405.4980* 15.

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

Candès, E. J., and Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9(6):717.

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM (JACM)* 58(3):11.

Candes, E. J.; Romberg, J. K.; and Tao, T. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59(8):1207–1223.

Cohen, M. B.; Musco, C.; and Musco, C. 2015. Ridge leverage scores for low-rank approximation. *arXiv preprint arXiv:1511.07263* 6.

Cohen, M. B.; Musco, C.; and Musco, C. 2017. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1758–1777. SIAM.

Dennis, Jr, J. E., and Moré, J. J. 1977. Quasi-Newton methods, motivation and theory. *SIAM review* 19(1):46–89.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2008. Relative-Error CUR Matrix Decompositions. *SIAM Journal on Matrix Analysis and Applications* 30(2):844–881.

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

Hsu, D.; Kakade, S.; and Zhang, T. 2014. Random Design Analysis of Ridge Regression. *Foundations of Computational Mathematics* 14(3).

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

Kasai, H., and Mishra, B. 2018. Inexact trust-region algorithms on Riemannian manifolds. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lee, J. D.; Sun, Y.; and Saunders, M. A. 2014. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization* 24(3):1420–1443.

Lin, H.; Mairal, J.; and Harchaoui, Z. 2015. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, 3384–3392.

Liu, D. C., and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45(1-3):503–528.

Liu, X.; Hsieh, C.-J.; Lee, J. D.; and Sun, Y. 2017. An inexact subsampled proximal Newton-type method for large-scale machine learning. *arXiv preprint arXiv:1708.08552*.

Mahajan, D.; Agrawal, N.; Keerthi, S. S.; Sellamanickam, S.; and Bottou, L. 2015. An efficient distributed learning algorithm based on effective local functional approximations. *Journal of Machine Learning Research* 16:1–36.

Mahoney, M. W. 2011. Randomized Algorithms for Matrices and Data. *Foundations and Trends in Machine Learning* 3(2):123–224.

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

Pilanci, M., and Wainwright, M. J. 2015. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research* 1–33.

Pilanci, M., and Wainwright, M. J. 2017. Newton Sketch: A Near Linear-Time Optimization Algorithm with Linear-Quadratic Convergence. *SIAM Journal on Optimization* 27(1):205–245.

Reddi, S. J.; Konevcnỳ, J.; Richtárik, P.; Póczós, B.; and Smola, A. 2016. AIDE: fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*.

Robbins, H., and Monro, S. 1985. A stochastic approximation method. In *Herbert Robbins Selected Papers*. Springer. 102–109.

Roosta, F.; Liu, Y.; Xu, P.; and Mahoney, M. W. 2018. Newton-MR: Newton's method without smoothness or convexity. *arXiv preprint arXiv:1810.00303*.

Roosta-Khorasani, F., and Mahoney, M. W. 2016. Sub-sampled Newton methods II: local convergence rates. *arXiv preprint arXiv:1601.04738*.

Schmidt, M.; LeRoux, N.; and Bach, F. 2017. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162(1–2):83–112.

Shalev Shwartz, S., and Ben David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shamir, O.; Srebro, N.; and Zhang, T. 2014. Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, 1000–1008.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Tripuraneni, N.; Stern, M.; Jin, C.; Regier, J.; and Jordan, M. I. 2018. Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Wang, S.; Roosta-Khorasani, F.; Xu, P.; and Mahoney, M. W. 2018. GIANT: Globally Improved Approximate Newton Method for Distributed Optimization. In *Thirty-Second Conference on Neural Information Processing Systems (NIPS)*.

Wang, S.; Gittens, A.; and Mahoney, M. W. 2018. Rketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *Journal of Machine Learning Research* 18(218):1–50.

Woodruff, D. P. 2014. Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends® in Theoretical Computer Science* 10(1–2):1–157.

Wright, S., and Nocedal, J. 1999. Numerical optimization. *Springer Science* 35(67–68):7.

Xu, P.; Yang, J.; Roosta Khorasani, F.; Ré, C.; and Mahoney, M. W. 2016. Sub-sampled newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, 3000–3008.

Xu, P.; Roosta Khorasan, F.; and Mahoney, M. W. 2017. Second-order optimization for non-convex machine learning: An empirical study. *arXiv preprint arXiv:1708.07827*.

Ye, H.; Luo, L.; and Zhang, Z. 2017. Approximate newton methods and their local convergence. In *International Conference on Machine Learning*, 3931–3939.

Yuan, X.-T., and Li, P. 2019. On convergence of distributed approximate Newton methods: Globalization, sharper bounds and beyond. *arXiv preprint arXiv:1908.02246*.

Zhang, Y., and Lin, X. 2015. DiSCO: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning*, 362–370.

Zhou, D.; Xu, P.; and Gu, Q. 2018. Stochastic variance-reduced cubic regularized Newton method. *arXiv preprint arXiv:1802.04796*.

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.