Proposal for Qualifying Exam

NAME OF STUDENT

Mar 17, 2020

Exam Committee

Committee Chairperson:

Prof. Jesús De Loera

Committee Members:

Prof. Krishna Balasubramanian

Prof. Miles Lopes

Prof. Wolfgang Polonik

Prof. Thomas Strohmer

Exam Logistics

Date:

Friday, May 8, 2020

Time:

10:00 AM - 13:00 PM

Location:

MSB 2240

Proposed Research Talk

Title. Estimating the Error of Randomized Newton Methods: A Bootstrap Approach

Abstract. Randomized Newton methods have recently become the focus of intense research activity in large-scale and distributed optimization. In general, these methods are based on a "computation-accuracy trade-off", which allows the user to gain scalability in exchange for error in the solution. However, the user does not know how much error is created by the randomized approximation, which can be detrimental in two ways: On one hand, the user may try to assess the unknown error with theoretical worst-case error bounds, but this approach is impractical when the bounds involve unknown constants, and it often leads to excessive computation. On the other hand, the user may select tuning parameters and stopping criteria in a heuristic manner, but this can lead to unreliable results. Motivated by these difficulties, we develop a bootstrap method for *directly estimating the unknown error*, which avoids both the excessive computation of the worst-case approach and the uncertainty of the heuristic approach. Also, we provide non-asymptotic theoretical guarantees to show that the error estimates are valid for several error metrics and algorithms (including GIANT and NEWTON SKETCH). Lastly, we show that the proposed method adds little cost to existing randomized Newton methods, and that it performs well in a range of experimental conditions.

1 Introduction

In recent years, there has been a surge of interest in using randomized approximations to accelerate Newton methods in large-scale and distributed optimization (e.g. Shamir et al., 2014; Erdogdu and Montanari, 2015; Zhang and Lin, 2015; Byrd et al., 2016; Pilanci and Wainwright, 2016; Reddi et al., 2016; Roosta-Khorasani and Mahoney, 2016; Xu et al., 2016; Pilanci and Wainwright, 2017; Wang et al., 2017, 2018; Dünner et al., 2018; Gupta et al., 2019; Li et al., 2019, among many others). At a high level, this rapid development of research has been driven by the fact that computing an inverse Hessian to machine precision can be very

costly or infeasible in large-scale problems. Instead, randomized approaches make it possible to overcome this challenge by exchanging some degree of accuracy in return for substantial reductions in both processing and communication costs. However, one of the common difficulties faced by users in applying randomized Newton methods is that they do not know how far a randomized Newton step might stray from an exact one.

To deal with the uncertainty in the quality of a randomized solution, we aim to develop a bootstrap-based method for estimating the errors of randomized Newton methods. In particular, this approach avoids the conservativeness of worst-case analysis by *directly estimating the actual error* of a given randomized solution. In addition, the bootstrap method provides the user with the flexibility to estimate error in terms of metrics for which worst-case bounds are unavailable, Next, in comparison to heuristic rules, this approach offers much more reliability by giving the user a systematic procedure that is supported by theoretical guarantees. Furthermore, the proposed bootstrap method is highly scalable due to its embarrassingly parallel structure, and it enjoys excellent numerical performance.

1.1 Background and setting

Consider the problem of minimizing an objective function of the form

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \varphi(a_i^\top w, b_i) + r(w), \tag{1}$$

where the functions $\varphi : \mathbb{R}^2 \to \mathbb{R}$ is convex and twice differentiable, $r : \mathbb{R}^d \to \mathbb{R}$ is twice differentiable, the vectors $a_1^\top, \ldots, a_n^\top \in \mathbb{R}^d$ are the rows of a matrix $A \in \mathbb{R}^{n\times d}$, the scalars $b_1, \ldots, b_n$ are the entries of a vector $b \in \mathbb{R}^n$, and $n \gg d$. Objective functions of the form (1) are ubiquitous in machine learning, where the points $\{(a_i, b_i)\}_{i=1}^n$ typically play the role of n observations, and f is viewed as a measure of empirical risk. Some of the most well known examples occur in the fitting of regularized *generalized linear models* and *support vector machines*, where φ corresponds to a loss function, and $r(w)$ is a regularization function of the form $r(w) = \gamma\|w\|_2^2$ for some parameter $\gamma > 0$ (cf. McCullagh, 2019; Chapelle, 2007). Another important class of examples arises in solving *linear programs* of the form $\min\{c^\top w \,|\, Aw \leqslant b\}$ by interior point methods, where $r(w) = c^\top w$ for some cost vector $c \in \mathbb{R}^d$, and φ corresponds to a logarithmic barrier function. (We refer to (Pilanci and Wainwright, 2017) for more detailed examples along these lines.)

Classical Netwon method. When a classical version of Newton's method is applied to minimize (1), each iterate $w_k \in \mathbb{R}^d$ is computed using both the gradient

$$g_k := \nabla f(w_k) = \frac{1}{n}\sum_{i=1}^{n} \partial_1\varphi(a_i^\top w_k, b_i)a_i + \nabla r(w_k)$$

as well as the Hessian $H_k := \nabla^2 f(w_k)$ given by

$$H_k = \frac{1}{n}\sum_{i=1}^{n} \partial_1^2\varphi(a_i^\top w_k, b_i)a_i a_i^\top + \nabla^2 r(w_k), \tag{2}$$

where ∂_1 is the partial derivative with respect to the first argument of φ. More specifically, if η_k is a step size parameter, then the update rule for the iterates is

$$w_{k+1} = w_k - \eta_k H_k^{-1} g_k.$$

However, in many cases, it is prohibitive to implement this update rule exactly, either because n is very large, or because the observations $\{(a_i, b_i)\}_{i=1}^n$ may be stored in a distributed manner, which can lead to high communication costs.

In order to overcome these bottlenecks, randomized Newton methods seek to compute efficient approximations to the inverse H_k^{-1}. In particular, these approximations tend to strongly leverage the fact that the function (1) has a Hessian that can be theoretically decomposed as

$$H_k = C_k^\top C_k + \nabla^2 r(w_k), \tag{3}$$

where $C_k \in \mathbb{R}^{n\times d}$ is a matrix given by

$$C_k := \tfrac{1}{\sqrt{n}} D_k A \quad \text{with} \quad D_k := \operatorname{diag}\{\sqrt{\partial_1^2 \varphi(a_i^\top w_k, b_i)}\}_{i=1}^n.$$

Below, we provide a very brief review to two well-known examples of such randomized algorithms called NEWTON SKETCH (Pilanci and Wainwright, 2017) and GIANT (Wang et al., 2018), since they will be the focus of our work later on.

The newton sketch algorithm. The core idea of the NEWTON SKETCH algorithm is to randomly transform the matrix C_k into a much shorter version $\tilde{C}_k \in \mathbb{R}^{t\times d}$ that can be handled more efficiently, where $t \ll n$. Commonly, the matrix $\tilde{C}_k$ is referred to as a "sketch" of C_k, and t is known as the "sketch size". In detail, the random transformation is implemented with a random "sketching matrix" $S_k \in \mathbb{R}^{t\times n}$ so that $\tilde{C}_k = S_k C_k$, and in turn, this leads to a sketched Hessian matrix defined as

$$\widetilde{H}_k := C_k^\top S_k^\top S_k C_k + \nabla^2 r(w_k).$$

Accordingly, this algorithm revises the classical Newton method by using the following randomized Newton step instead,

$$w_{k+1} = w_k - \eta_k (\widetilde{H}_k)^{-1} g_k. \tag{4}$$

In order to ensure that $\tilde{H}_k$ provides an effective approximation to H_k, the sketching matrix S_k is typically generated so that it has i.i.d. rows and satisfies the relation $\mathbb{E}[S_k^\top S_k] = I_n$. For example, when S_k is a *uniform sampling matrix*, the rows of S_k are generated as i.i.d. samples from the uniform distribution on the set $\{\sqrt{n}e_1, \ldots, \sqrt{n}e_n\} \subset \mathbb{R}^n$, where $e_1, \ldots, e_n$ are the canonical basis vectors.

The giant algorithm. When data are stored on a distributed system, communication cost between different machines (workers) is often of paramount importance. As a way to avoid the high-communication costs that arise when computing an exact Newton step in this setting, the GIANT algorithm uses an approximation to H_k^{-1} derived from the harmonic mean of local Hessian matrices.

To be more specific, suppose random samples from $\{(a_i, b_i)\}_{i=1}^n$ are evenly distributed across m different workers, and the j-th worker holds data points with indices in set $\mathcal{I}_j$. Then, the j-th local Hessian matrix at the k-th iteration is defined as

$$\widetilde{H}_{k,j} := \tfrac{1}{|\mathcal{I}_j|} \textstyle\sum_{i\in\mathcal{I}_j} \partial_1^2 \varphi(a_i^\top w_k, b_i) a_i a_i^\top + \nabla^2 r(w_k), \tag{5}$$

and the Newton update takes the form

$$\begin{aligned} w_{k+1} &= w_k - \eta_k \tfrac{1}{m} \textstyle\sum_{j=1}^m (\widetilde{H}_{k,j})^{-1} g_k \\ &= w_k - \eta_k (\widetilde{H}_k)^{-1} g_k, \end{aligned} \tag{6}$$

where $\widetilde{H}_k := \big(\frac{1}{m}\sum_{j=1}^m (\widetilde{H}_{k,j})^{-1}\big)^{-1}$ is the approximate Hessian matrix.

1.2 Problem formulation

In order to study the algorithmic error of randomized Newton methods, our work will focus entirely on the randomness that comes from within the algorithms, and we will always treat the points $\{a_i, b_i\}_{i=1}^n$ and the function f as being deterministic. From this perspective, it is important to clarify that the iterates w_k of such algorithms are random vectors, but the exact optimal solution

$$w_{\text{opt}} = \operatorname*{argmin}_{w\in\mathbb{R}^d} f(w)$$

is deterministic.

Estimating error with respect to Newton step. To measure the quality of the iterate w_k, we may consider the (random) error in Newton step

$$\epsilon_k = \rho(\widetilde{\Delta}_k, \Delta_k), \tag{7}$$

where $\rho(\cdot, \cdot)$ is a generic measure of error, and we write

$$\Delta_k := H_k^{-1} g_k \qquad \text{and} \qquad \widetilde{\Delta}_k := (\widetilde{H}_k)^{-1} g_k \tag{8}$$

for the exact and sketched Newton steps. For example, we can take $\rho(w', w)$ to be the absolute error in any norm $\|\cdot\|_\diamond$, i.e. $\rho(w', w) = \|w' - w\|_\diamond$, or the relative error $\rho(w', w) = \|w' - w\|_\diamond / \|w\|_\diamond$. The error in Newton step is of particular interest for functions that are locally quadratic near optimal solution, since for quadratic functions, the exact Newton method will converge in a single step.

Due to the fact that the error ϵ_k of Newton step is a random variable, it is of interest to study its $(1-\alpha)$-quantile, which is defined as the *tighest possible* upper bound on ϵ_k that holds with probability at least $1 - \alpha$,

$$q_{\alpha,k} = \inf \Big\{ q \in (-\infty, \infty) \,\Big|\, \mathbb{P}\big(\epsilon_k \leqslant q\big) \geqslant 1 - \alpha \Big\}.$$

Since the quantile $q_{\alpha,k}$ is unknown in practice, we aim to construct an estimate $\widehat{q}_{\alpha,k}$, which is intended to satisfy the bound

$$\epsilon_k \leqslant \widehat{q}_{\alpha,k} \tag{9}$$

with probability nearly equal to $1 - \alpha$.

Estimating error with respect to Newton decrement. Another commonly used error metric is the optimality gap $f(w_k) - f(w_{\text{opt}})$. To derive a bound on the optimality gap, it is convenient to consider the *squared Newton decrement*

$$\delta_k^2 := g_k^\top H_k^{-1} g_k. \tag{10}$$

This quantity has special significance when f is a convex function that satisfies the condition of *self-concordance* — which commonly arises in the context of interior point methods (cf. Nesterov and Nemirovskii, 1994). In fact, some of the most widely studied instances of the function (1) are known to be self-concordant, including those arising from ridge regression, regularized logistic, and smoothed hinge loss functions (cf. Zhang and Lin, 2015).

For a function f that is convex and self-concordant, it is a classical fact that if w_k is any point in the function's domain, then the optimality gap is bounded according to

$$f(w_k) - f(w_{\text{opt}}) \;\leqslant\; \delta_k^2, \tag{11}$$

provided that $\delta_k \leqslant 0.68$ (Nesterov and Nemirovskii, 1994). However, because the exact quantity δ_k^2 is unknown, it is of interest to measure the error of the approximate squared decrement $\widetilde{\delta}_k^2 := g_k^\top \tilde{H}_k^{-1} g_k$, namely

$$\varepsilon_k = \varrho(\widetilde{\delta}_k^2, \delta_k^2), \tag{12}$$

where $\varrho(\cdot, \cdot)$ is an error measure of our choice. Also, by analogy with the earlier definition of $q_{\alpha,k}$, the $(1-\alpha)$-quantile of ε_k is defined as

$$\mathtt{q}_{\alpha,k} = \inf\{q \in (-\infty, \infty) | \mathbb{P}(\varepsilon_k \leqslant q) \geqslant 1 - \alpha\}.$$

Furthermore, since this parameter is unknown, we seek to construct an estimate $\widehat{\mathtt{q}}_{\alpha,k}$ such that the following bound holds with probability nearly equal to $1 - \alpha$,

$$\varepsilon_k \leqslant \widehat{\mathtt{q}}_{\alpha,k}. \tag{13}$$

In turn, this will provide a high probability bound on the optimality gap. For example, when ϱ is chosen to be the relative error $\varrho(\widetilde{x}, x) = |\widetilde{x} - x|/|x|$, the estimate $\widehat{\mathtt{q}}_{\alpha,k}$ will be expected to satisfy

$$f(w_k) - f(w_{\text{opt}}) \;\leqslant\; \tilde{\delta}_k^2 \, (1 - \widehat{\mathtt{q}}_{\alpha,k})^{-1}$$

with probability at least $1-\alpha$.

2 Methods

In this section, we describe two methods for constructing estimates $\widehat{q}_{\alpha,k}$ and $\widehat{\mathtt{q}}_{\alpha,k}$ satisfying the conditions (9) and (13) respectively.

2.1 Bootstrap Error Estimation for newton sketch

Since in practice we are not computing the exact Hessian, the actual distribution of ϵ_k remains unknown. However, if we can generate some bootstrap samples of Newton step $\widetilde{\Delta}_k^*$ that fluctuate around $\widetilde{\Delta}_k$ in a similar way as $\widetilde{\Delta}_k$ fluctuate around Δ_k, then the empirical $(1-\alpha)$-quantile of the error $\rho(\widetilde{\Delta}_k^*, \widetilde{\Delta}_k)$ should be close to the quantile for the actual error ϵ_k, and similarly the empirical quantile of $\varrho(\widetilde{\delta}_k^{*2}, \widetilde{\delta}_k^2)$ should be a good estimator of the true quantile for ε_k. Here $\widetilde{\delta}_k^{*2}$ are the bootstrap samples of squared Newton decrement, defined as $\widetilde{\delta}_k^{*2} = g_k^\top \widetilde{\Delta}_k^*$.

The proposed method is based on the preceding observation. To generate the relevant bootstrap samples, we generate a matrix $S_k^* \in \mathbb{R}^{t\times n}$ by sampling t rows uniformly (i.e. with replacement) from S_k. In turn, a bootstrap sample of the Hessian can be constructed as $\widetilde{H}_k^* = C_k^\top S_k^{*\top} S_k^* C_k + \nabla^2 r(w_k)$, and then the bootstrap samples of the Newton step and squared decrement can be computed accordingly.

The exact details are listed below in Algorithm 1.

Algorithm 1: Error estimation for NEWTON SKETCH

Input: The current iterate $w_k \in \mathbb{R}^d$, the sketch $\tilde{C}_k \in \mathbb{R}^{t\times d}$, the sketched Newton step $\tilde{\Delta}_k \in \mathbb{R}^{d\times 1}$, sketched squared decrement $\tilde{\delta}_k^2$, as well as the current gradient $g_k \in \mathbb{R}^d$.

for $b = 1$ **to** B **do in parallel**

- Construct a matrix $\tilde{C}_k^* \in \mathbb{R}^{t\times d}$ whose rows are sampled uniformly from the rows of $\tilde{C}_k$.
- Compute the following in succession:

$$\tilde{H}_k^* := (\tilde{C}_k^*)^\top (\tilde{C}_k^*) + \nabla^2 r(w_k)$$
$$\tilde{\Delta}_k^* := (\tilde{H}_k^*)^{-1} g_k$$
$$\widetilde{\delta}_k^{*2} := g_k^\top \tilde{\Delta}_k^*$$
$$\tilde{\epsilon}_{k,b}^* := \rho(\tilde{\Delta}_k^*, \tilde{\Delta}_k)$$
$$\tilde{\varepsilon}_{k,b}^* := \varrho(\tilde{\delta}_k^{*2}, \tilde{\delta}_k^2)$$

Return: $\widehat{q}_{\alpha,k} := \text{quantile}(\epsilon_{k,1}^*, \ldots, \epsilon_{k,B}^*; 1-\alpha)$
$\widehat{\mathfrak{q}}_{\alpha,k} := \text{quantile}(\varepsilon_{k,1}^*, \ldots, \varepsilon_{k,B}^*; 1-\alpha)$

2.2 Bootstrap Error Estimation for giant

We briefly recall the setting of the GIANT algorithm, where we have m workers, and the j-th worker holds $t = n/m$ data points drawn from $(a_i, b_i)_{i=1}^n$, indexed by $\mathcal{I}_j$.

To construct a GIANT step, each worker compute a locally approximated Newton step

$$\widetilde{\Delta}_{k,j} = \widetilde{H}_{k,j}^{-1} g_k,$$

where $\widetilde{H}_{k,j}$ is defined as in (5). Note that the construction of $\widetilde{H}_{k,j}$ only requires access to local samples. The globally improved approximated Newton step (GIANT) $\widetilde{\Delta}_k$ is then computed by aggregating the local steps $\widetilde{\Delta}_{k,j}$, and the GIANT steps will be used to update the current iterate.

Since the data points on each worker are randomly drawn from $\{(a_i, b_i)\}_{i=1}^n$Wang et al. (2018), Algorithm 1 can naturally be extended to accommodate the distributed setting with uniform sampling. That is, if we want to generate bootstrap step samples that fluctuate around GIANT step in a similar way as GIANT step fluctuates around the exact Newton step Δ_k, then we can follow the process similarly as in Algorithm 1, to generate local bootstrap Newton steps on each worker, and aggregate them to obtain a global bootstrap

Newton step sample. The process for generating the squared decrement samples are similar, and the exact details are listed in Algorithm 2.

Algorithm 2: Error estimation for GIANT

Input: The current iterate $w_k \in \mathbb{R}^d$, index sets $\mathcal{I}_1, \ldots, \mathcal{I}_m$, GIANT step $\tilde{\Delta}_k$, approximate squared decrement $\widetilde{\delta}_k^2$, as well as the gradient g_k.

for $j = 1$ **to** m **do in parallel**

  for $b = 1$ **to** B **do in parallel**

  - Draw t indices uniformly from $\mathcal{I}_j$ to form $\mathcal{I}^*_{j,b}$.
  - Compute$\tilde{H}^*_{k,j,b} := \frac{1}{t}\sum_{i \in \mathcal{I}^*_{j,b}} \partial_1^2 \varphi(a_i^\top w_k, b_i) a_i a_i^\top + \nabla^2 r(w_k)$..
  - Compute $\tilde{\Delta}^*_{k,j,b} := (\tilde{H}^*_{k,j,b})^{-1} g_k$.
  - Compute $\tilde{\delta}^{*2}_{k,j,b} := g_k^\top \tilde{\Delta}^*_{k,j,b}$.

- Aggregate local $*$-steps $\tilde{\Delta}^*_{k,b} := \frac{1}{m}\sum_{j=1}^m \tilde{\Delta}^*_{k,j,b}$.
- Aggregate local $*$-decrements $\tilde{\delta}^{*2}_{k,b} := \frac{1}{m}\sum_{j=1}^m \tilde{\delta}^{*2}_{k,j,b}$.
- Compute $*$-step error $\epsilon^*_{k,b} := \rho(\tilde{\Delta}^*_{k,b}, \tilde{\Delta}_k)$.
- Compute $*$-decrement error $\varepsilon^*_{k,b} := \varrho(\tilde{\delta}^{*2}_{k,b}, \tilde{\delta}^2_k)$.

Return: $\widehat{q}_{\alpha,k} := \text{quantile}(\epsilon^*_{k,1}, \ldots, \epsilon^*_{k,B}; 1-\alpha)$,
$\widehat{\mathsf{q}}_{\alpha,k} := \text{quantile}(\varepsilon^*_{k,1}, \ldots, \varepsilon^*_{k,B}; 1-\alpha)$.

3 Main Theoretical Results

In this section, we analyze the proposed method for minimizing functions of the form (1), with $r(w) = \frac{\gamma}{2}\|w\|_2^2$ for some regularization parameter $\gamma > 0$. For the sake of brevity, our theoretical results will focus on the particular relative error metric $\varrho(x', x) = |x' - x|/|x|$ for the Newton decrement.

3.1 Theoretical Setup

Consider a distributed setting where there are m workers that have $t = n/m$ data points obtained as uniform samples from $\{(a_i, b_i)\}_{i=1}^n$ (which is the main setting considered by the authors of GIANT). Also, in order to unify the presentation of our results, it is helpful to note that when $m = 1$, this setting allows NEWTON SKETCH to be analyzed with uniform sampling matrices S_k and a sketch size of t. In addition, our proofs can be extended in a straightforward way to accommodate other types of sketching matrices for NEWTON SKETCH, but we omit such details to keep our results more concise.

Notation and definitions. For the regularization function $r(w) = \frac{\gamma}{2}\|w\|_2^2$, recall that the Hessian H_k can be decomposed as

$$H_k = C_k^\top C_k + \gamma I_d. \tag{14}$$

We define the i-th *ridge leverage score* as

$$\ell^\gamma_{i,k} := \left[C_k (C_k^\top C_k + \gamma I_d)^{-1} C_k^\top\right]_{ii} \tag{15}$$

When $\gamma = 0$, this coincides with the standard leverage score. Next, the *effective dimension* d_k^γ is defined as as

$$d_k^\gamma := \ell_{1,k}^\gamma + \cdots + \ell_{n,k}^\gamma = \sum_{j=1}^d \frac{\sigma_j^2(C_k)}{\sigma_j^2(C_k)+\gamma}, \tag{16}$$

which can be much smaller than d when C_k has only a few dominant singular values (cf. Li et al., 2019). We also use $\mu_k^\gamma = \mu_k^\gamma(C_k)$ to refer to the ridge coherence, defined as

$$\mu_k^\gamma := \max_{1 \leqslant i \leqslant n} \Big(\frac{n\ell_{i,k}^\gamma}{d_k^\gamma} \vee 1 \Big). \tag{17}$$

In the case when γ is set to 0, the quantity μ_k^γ reduces to the ordinary matrix coherence (cf. Candès and Recht, 2009), but it should be noted that γ will always assumed to be positive in our results. Intuitively μ_k^γ measures how evenly information is spread among the rows of matrix C_k, and in many natural settings, this quantity is $\mathcal{O}(1)$ or $\mathcal{O}(\log(n))$. (See also Proposition 1 later on.)

Below, the theoretical assumptions for our results are stated in terms of a random vector $s \in \mathbb{R}^n$ that is sampled from the uniform distribution on the set $\{\sqrt{n}e_1, \ldots, \sqrt{n}e_n\} \subset \mathbb{R}^n$.

Assumption 1. There is an absolute constant $c_1 > 0$ such that the following bound holds for any $k \geqslant 1$ when $g_k \neq 0$,

$$\operatorname{var}\Big(\tfrac{1}{\delta_k^2} g_k^\top H_k^{-1} C_k^\top s s^\top C_k H_k^{-1} g_k \Big) \geqslant c_1. \tag{18}$$

Remark. In Section 3.2, we show that the above inequalities in Assumptions 1 holds "generically", in the sense that if the Q factor in the QR-factorization of C_k is drawn uniformly at random, then (18) holds with high probability.

The following theorems provide theoretical guarantee on the performance of the proposed method for estimating the error in squared Newton decrement.

Theorem 3.1. Suppose that Assumptions 1 holds, and let $\widehat{\mathsf{q}}_{\alpha,k}$ denote the output of Algorithm 1 or 2. Then, there exists absolute constant $c > 0$, such that the bound

$$\mathbb{P}\Big(\tfrac{|\tilde{\delta}_k^2 - \delta_k^2|}{\delta_k^2} \leqslant \widehat{\mathsf{q}}_{\alpha,k} \Big) \;\geqslant\; 1 - \alpha - c\,\omega$$

holds for some positive number ω satisfying

$$\omega \leqslant \tfrac{(\mu_k^\gamma d_k^\gamma)^3 \log(n)^c}{\sqrt{t}} + \tfrac{\sqrt{\log(B)}}{\sqrt{B}}.$$

Remarks. The main idea in our analysis is to approximate the distribution of error variable on each worker by a Gaussian distribution. However, if we use the existing techniques for analyzing the error on a single worker as inLopes et al. (2018); Fang et al. (2018), then in the distributed setting, where we need to aggregating the results over m workers, the bias in each worker will also be accumulated. In other words, if we follow the existing literature, the error term on the distributed system will be greater than on the single worker setting. This does not match the experimental results. To address this issue, we further take into account the bias on each worker, and show that the bootstrap method we proposed in Algorithm 1,2 can correct the bias regardless of the number of workers, as is implied by our main theorem.

3.2 Validating the Assumptions for Generic Matrices

The collection of matrices in $\mathbb{R}^{n\times d}$ with orthornormal columns, denoted as $\mathsf{Stief}(n,d)$, is known to possess a natural uniform probability distribution Meckes (2019). From a conceptual point of view, a random matrix $Q \in \mathsf{Stief}(n,d)$ generated from this distribution may be regarded as "generic". Accordingly, we can investigate whether or not our theoretical assumption holds generically by checking if they are likely to hold when the Q factor in the QR-decomposition of C_k is drawn from this distribution. In this way, the following proposition provides a source of validation for Assumptions 1. In addition, the inequality (19) below shows that the quantity $\mu_k^\gamma d_k^\gamma$ is of moderate size in this situation.

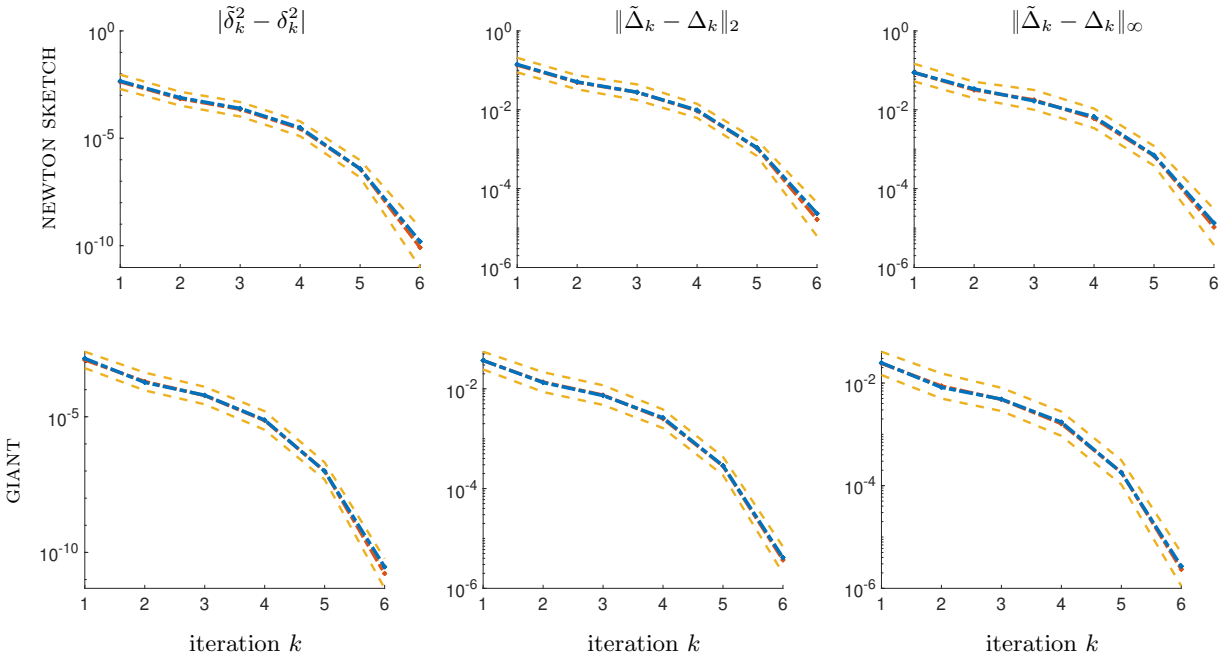

Figure 1: Numerical results on dataset SUSY ($n = 5{,}000{,}000, d = 18$). The plots illustrate the performance of Algorithms 1 and 2 in the task of estimating the quantiles of the three errors $|\tilde{\delta}_k^2 - \delta_k^2|$, $\|\tilde{\Delta}_k - \Delta_k\|_2$, and $\|\tilde{\Delta}_k - \Delta_k\|_\infty$. The blue curves are the empirical estimation of the 0.95-quantile of the errors, and are regarded as the ideal benchmarks. The red curve is the averaged bootstrap estimation of the quantile, with the yellow curves being three standard deviation away.

Proposition 1. Fix any integer $k \geqslant 1$, and let $R \in \mathbb{R}^{d\times d}$ be any fixed upper-triangular matrix satisfying

$$\tfrac{1}{c} \leqslant \sigma_{\min}(R) \leqslant \sigma_{\max}(R) \leqslant c$$

for some absolute constant $c > 0$. Next, let $Q \in \mathbb{R}^{n\times d}$ be a random matrix drawn from the uniform distribution on $\mathsf{Stief}(n, d)$, and suppose that the matrix C_k in (14) is equal to QR. Under these conditions, there exists an absolute constant $c_0 > 0$, such that if $n \geqslant c_0 d^2$, then the inequalities (18) in Assumptions 1 holds with probability at least $1 - \frac{1}{n}c_0(d \vee \log n)^2$. Furthermore, the inequality

$$\mu_k^\gamma d_k^\gamma \leqslant c_0(d_k^\gamma \vee \log n) \tag{19}$$

holds with probability at least $1 - c_0/n$.

Remarks. Since the columns of a random matrix Q drawn uniformly on $\mathsf{Stief}(n, d)$ are *not independent*, it is necessary in the proof of this result to make use of non-asymptotic tools that can allow for such dependence. Specifically, the proof hinges on the fact that if a function $\psi : \mathsf{Stief}(n, d) \to \mathbb{R}$ is Lipschitz with respect to the Frobenius norm, then the random variable $\psi(Q)$ has strong concentration properties (cf. Milman and Schechtman, 2009, p.29).

4 Experiments

We conducted a collection of experiments showing how the proposed error estimation methods can be applied to NEWTON SKETCH and GIANT in the context of ℓ_2-regularized logistic regression. Specifically, the objective function we are considering has the form

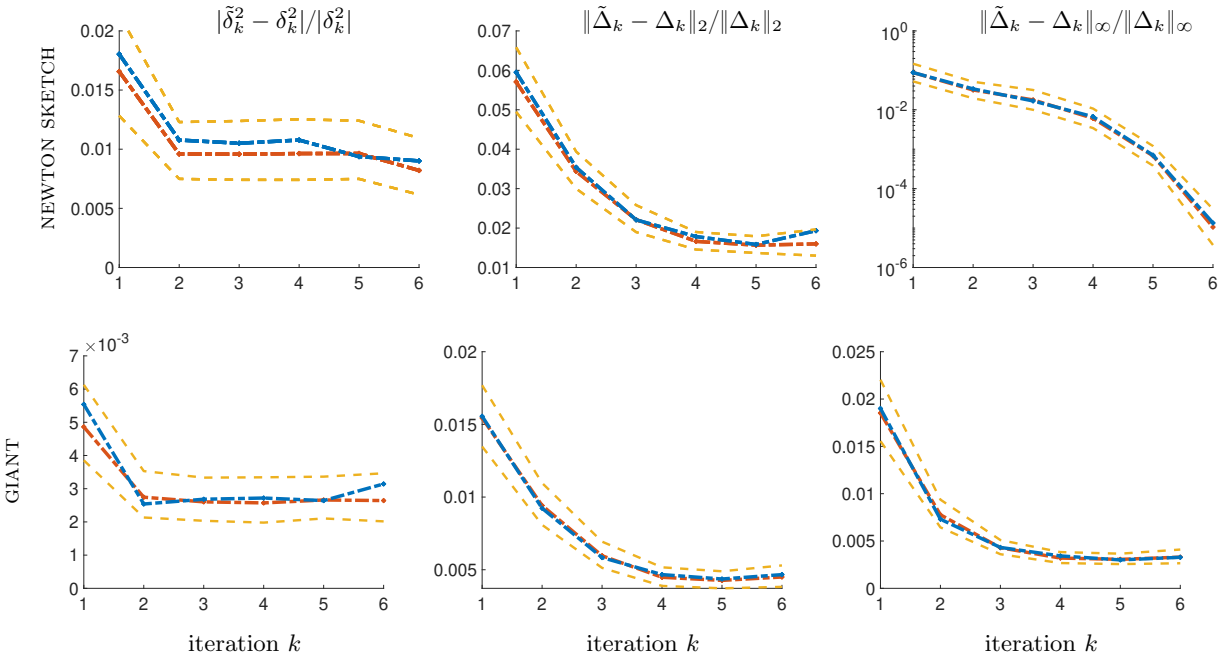

Figure 2: Numerical results on dataset SUSY ($n = 5{,}000{,}000, d = 18$). The plots illustrate the performance of Algorithms 1 and 2 in the task of estimating the quantiles of the three relative errors $|\tilde{\delta}_k^2 - \delta_k^2|/|\delta_k^2|$, $\|\tilde{\Delta}_k - \Delta_k\|/\|\Delta_k\|_2$, and $\|\tilde{\Delta}_k - \Delta_k\|_\infty/\|\Delta_k\|_\infty$. The blue curves are the empirical estimation of the 0.95-quantile of the errors, and are regarded as the ideal benchmarks. The red curve is the averaged bootstrap estimation of the quantile, with the yellow curves being three standard deviation away.

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-(a_i^\top w)b_i)) + \frac{\gamma}{2}\|w\|_2^2,$$

where the observations satisfy $(a_i, b_i) \in \mathbb{R}^d \times \{\pm 1\}$ for all $i \in \{1, \ldots, n\}$.

Data and parameter settings. For illustration we present the result for the regression dataset: SUSY ($n = 5{,}000{,}000, d = 18$), which can be obtained from LIBSVM Chang and Lin (2011). We set the regularization parameter to be $\gamma = 10^{-3}$, and the number of bootstrap samples to be $B = 12$. We determine the step length η_k in each iteration by Armijo line search, so that

$$f(w_k + \eta_k \widetilde{\Delta}_k) \leqslant f(w_k) + \eta_k \beta \langle \widetilde{\Delta}_k, g_k \rangle.$$

In our experiments, we set the control parameter $\beta = 0.1$, and η_k is chosen from $\{2^0, 2^{-1}, \ldots, 2^{-10}\}$.

We estimated the quantiles of both the relative error and the absolute error in three error metrics: the error in squared Newton decrement, in ℓ_2-norm of the Newton step, and in ℓ_∞-norm of the Newton step. The results are shown in Figure 1, 2.

Below, we give the detailed description of the experiments for Algorithm 2 (GIANT) respectively. The experiments for Algorithm 1 (NEWTON SKETCH) is conducted in a similar way.

Experiments for giant. The experiments for GIANT algorithm are conducted in a similar way. We ran the GIANT algorithm for 300 times, each time with 6 iterations using the update form as in (6). We randomly sampled t data points for each of the m workers before each run, and the data points stayed unchanged on each worker throughout the iterations. We chose m to be $m = 32$ for all datasets in correspondence with the sketch size in NEWTON SKETCH. The error variables were computed at each iteration, giving 300 realizations of each error variable, and the 0.95-quantiles of the 300 recorded errors at iteration $k = 1 \ldots, 6$ were then

computed and were considered as the ideal empirical estimation of the true quantiles. As before we plot them in blue. We also ran Algorithm 2 to obtain a bootstrap estimation of the quantiles. The bootstrap estimations are plotted against the empirical estimation, see figure 1,2.

5 Future work

- We are working on deriving and improving the theoretical bound for the error in Newton step as well. Also, similar techniques could be used to extend the results to other sketching method, such as length square sampling, Gaussian Projection, subsampled randomized Hadamard transform, etc.
- We would like to adapt bootstrap methods for posteriori error estimation of randomized algorithms in other regimes, for example, Schatten norm estimation, low rank matrix approximation, sparse graph approximation, or zero-th order optimization. Try to propose algorithms that effectively estimate the error while keeping the computation (and possibly communication) cost relatively low, and provide non-asymptotic theoretical guarantees for the proposed methods.

References

Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

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

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Olivier Chapelle. Training a support vector machine in the primal. *Neural computation*, 19(5):1155–1178, 2007.

Germund Dahlquist and Ake Bjorck. *Numerical Methods in Scientific Computing: Volume 1*, volume 103. Siam, 2008.

Celestine Dünner, Aurelien Lucchi, Matilde Gargiani, An Bian, Thomas Hofmann, and Martin Jaggi. A distributed second-order algorithm you can trust. *arXiv:1806.07569*, 2018.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. *arXiv:1508.02810*, 2015.

Yixin Fang, Jinfeng Xu, and Lei Yang. Online bootstrap confidence intervals for the stochastic gradient descent estimator. *The Journal of Machine Learning Research*, 19(1):3053–3073, 2018.

James E Gentle. *Computational statistics*, volume 308. Springer, 2009.

Vipul Gupta, Swanand Kadhe, Thomas Courtade, Michael W Mahoney, and Kannan Ramchandran. Oversketched newton: Fast convex optimization for serverless systems. *arXiv:1903.08857*, 2019.

Xiang Li, Shusen Wang, and Zhihua Zhang. Do subsampled newton methods work for high-dimensional data? *arXiv:1902.04952*, 2019.

M. E. Lopes, S. Wang, and M. W. Mahoney. Error estimation for randomized least-squares algorithms via the bootstrap. *arXiv:1803.08021*, 2018.

Peter McCullagh. *Generalized linear models*. Routledge, 2019.

Elizabeth S Meckes. *The Random Matrix Theory of the Classical Compact Groups*, volume 218. Cambridge University Press, 2019.

Vitali D Milman and Gideon Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaces: Isoperimetric Inequalities in Riemannian Manifolds*, volume 1200. Springer, 2009.

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

Mert Pilanci and Martin J Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *The Journal of Machine Learning Research*, 17(1):1842–1879, 2016.

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.

Sashank J Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. AIDE: Fast and communication efficient distributed optimization. *arXiv:1608.06879*, 2016.

Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods I: globally convergent algorithms. *arXiv:1601.04737*, 2016.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008, 2014.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Jialei Wang, Jason D Lee, Mehrdad Mahdavi, Mladen Kolar, Nathan Srebro, et al. Sketching meets random projection in the dual: A provable recovery algorithm for big and high-dimensional data. *Electronic Journal of Statistics*, 11(2):4896–4944, 2017.

Shusen Wang, Farbod Roosta-Khorasani, Peng Xu, and Michael W Mahoney. Giant: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, pages 2332–2342, 2018.

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

Yuchen Zhang and Xiao Lin. DiSCO: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370, 2015.

Proposed Exam Syllabus

Topic #1: Numerical Methods

Reference:Dahlquist and Bjorck (2008)

- Float point arithmetic, error propagation
- Matrix Computations
  - LU factorization, QR decomposition, Singular value decomposition
  - Krylov Subspace, Conjugate gradient method

  - Power iterations, Lanczos process
  - Randomized Matrix Multiplication
  - Sketching and randomized least square problem
- Numerical Optimization
  - Gradient Descent, Secant Method, Newton's Method
  - Subgradient Method, Projected Gradient Descent, Alternating Direction Method of Multipliers

Topic #2: Probability Theory

Reference: Durrett (2019)

- Measure theory
  - Random variables
  - Integration and Expectation
- Law of large numbers
  - Borel-Cantelli lemmas
  - Weak and strong law of large numbers
  - Glivenko–Cantelli theorem
- Central limit theorem
  - Generating functions and moments
  - Central limit theorems

Topic #3: High Dimensional Statistics

Reference: Vershynin (2018)

- Tail and concentration bounds
  - Chernoff's inequality, Hoeffiding's inequality, Bernstein's inequality
  - Sub-Gaussian, sub-exponential distributions
  - Concentration of Lipschitz functions on the sphere, Gaussian concentration inequality
  - Johnson-Linderstrauss lemma
- Random Process
  - Covering numbers and packing numbers
  - Gaussian and Rademacher complexity
  - Slepian's inequality, Sudakov-Fernique's inequality
  - Sudakov's minoration inequality
  - Chaining, Dudley's inequality

Topic #4: Computational Statistics

Reference: Gentle (2009)

- Expectation Maximization Algorithm
- Sampling techniques
  - Inverse transform, Rejection sampling
  - Markov Chain Monte Carlo