Distributed Newton Can Communicate Less and Resist Byzantine Workers

Anonymous Author(s)

Abstract

We develop a distributed second order optimization algorithm that is communication-efficient as well as robust against Byzantine failures of the worker machines. We propose COMRADE (COMunication-efficient and Robust Approximate Distributed nEwton), an iterative second order algorithm, where the worker machines communicate *only once* per iteration with the center machine. This is in sharp contrast with the state-of-the-art distributed second order algorithms like GIANT [30] and DINGO[6], where the worker machines send (functions of) local gradient and Hessian sequentially; thus ending up communicating twice with the center machine per iteration. Moreover, we show that the worker machines can further compress the local information before sending it to the center. In addition, we employ a simple norm based thresholding rule to filter-out the Byzantine worker machines. We establish the linear-quadratic rate of convergence of COMRADE and establish that the communication savings and Byzantine resilience result in only a small statistical error rate for arbitrary convex loss functions. To the best of our knowledge, this is the first work that addresses the issue of Byzantine resilience in second order distributed optimization. Furthermore, we validate our theoretical results with extensive experiments on synthetic and benchmark LIBSVM [4] data-sets and demonstrate convergence guarantees.

1 Introduction

In modern data-intensive applications like image recognition, conversational AI and recommendation systems, the size of training datasets has grown in such proportions that distributed computing have become an integral part of machine learning. To this end, a fairly common distributed learning framework, namely *data parallelism*, distributes the (huge) data-sets over multiple *worker machines* to exploit the power of parallel computing. In many applications, such as Federated Learning [17], data is stored in users' personal devices and judicious exploitation of the on-device machine intelligence can speed up computation. Usually, in a distributed learning framework, computation (such as processing, training) happens in the worker machines and the local results are communicated to a *center machine* (ex., a parameter server). The center machine updates the model parameters by properly aggregating the local results.

Such distributed frameworks face the following two fundamental challenges: First, the parallelism gains are often bottle-necked by the heavy communication overheads between worker and the center machines. This issue is further exacerbated where large clusters of worker machines are used for modern deep learning applications using models with millions of parameters (NLP models, such as BERT [9], may have well over 100 million parameters). Furthermore, in Federated Learning, this uplink cost is tied to the user's upload bandwidth. Second, the worker machines might be susceptible to errors owing to data crashes, software or hardware bugs, stalled computation or even malicious and co-ordinated attacks. This inherent unpredictable (and potentially adversarial) nature of worker machines is typically modeled as Byzantine failures. As shown in [18], Byzantine behavior a single worker machine can be fatal to the learning algorithm.

Both these challenges, communication efficiency and Byzantine-robustness, have been addressed in a significant number of recent works, albeit mostly separately. For communication efficiency, several recent works [28, 27, 2, 13, 1, 31, 16] use quantization or sparsification schemes to compress the

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

message sent by the worker machines to the center machine. An alternative, and perhaps more natural way to reduce the communication cost (via reducing the number of iterations) is to use second order optimization algorithms; which are known to converge much faster than their first order counterparts. Indeed, a handful of algorithms has been developed using this philosophy, such as DANE [24], DISCO [34], GIANT [30] , DINGO [6], Newton-MR [23], INEXACT DANE and AIDE [22]. On the other hand, the problem of developing Byzantine-robust distributed algorithms has also been considered recently (see [26, 12, 5, 32, 33, 14, 3]). However, *all* of these papers analyze different variations of the gradient descent, the standard first order optimization algorithm.

In this work, we propose COMRADE, a distributed approximate Newton-type algorithm that communicates less and is resilient to Byzantine workers. Specifically, we consider a distributed setup with m worker machines and one center machine. The goal is to minimize a regularized convex loss $f : \mathbb{R}^d \to \mathbb{R}$, which is additive over the available data points. Furthermore, we assume that α fraction of the worker machines are Byzantine, where $\alpha \in [0, 1/2)$. We assume that Byzantine workers can send any arbitrary values to the center machine. In addition, they may completely know the learning algorithm and are allowed to collude with each other. To the best of our knowledge, this is the first paper that addresses the problem of Byzantine resilience in second order optimization.

In our proposed algorithm, the worker machines communicate *only once* per iteration with the center machine. This is in sharp contrast with the state-of-the-art distributed second order algorithms (like GIANT [30], DINGO [6], Determinantal Averaging [8]), which sequentially estimates functions of local gradients and Hessians and communicate them with the center machine. In this way, they end up communicating twice per iteration with the center machine. We show that this sequential estimation is redundant. Instead, in COMRADE, the worker machines only send a d dimensional vector, the product of the inverse of local Hessian and the local gradient. Via sketching arguments, we show that the empirical mean of the product of local Hessian inverse and local gradient is close to the global Hessian inverse and gradient product, and thus just sending the above-mentioned product is sufficient to ensure convergence. Hence, in this way, we save $\mathcal{O}(d)$ bits of communication per iteration. Furthermore, in Section 5, we argue that, in order to cut down further communication, the worker machines can even compress the local Hessian inverse and gradient product. Specifically, we use a (generic) ρ-approximate compressor ([16]) for this, that encompasses sign-based compressors like QSGD [1] and top_k sparsification [25].

For Byzantine resilience, COMRADE employs a simple thresholding policy on the norms of the local Hessian inverse and local gradient product. Note that norm-based thresholding is computationally much simpler in comparison to existing co-ordinate wise median or trimmed mean ([32]) algorithms. Since the norm of the Hessian-inverse and gradient product determines the *amount* of movement for Newton-type algorithms, this norm corresponds to a natural metric for identifying and filtering out Byzantine workers.

Our Contributions: We propose a communication efficient Newton-type algorithm that is robust to Byzantine worker machines. Our proposed algorithm, COMRADE takes as input the local Hessian inverse and gradient product (or a compressed version of it) from the worker machines, and performs a simple thresholding operation on the norm of the said vector to discard $\beta > \alpha$ fraction of workers having largest norm values. We prove the linear-quadratic rate of convergence of our proposed algorithm for strongly convex loss functions. In particular, suppose there are m worker machines, each containing s data points; and let $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$, where $\mathbf{w}_t$ is the t-th iterate of COMRADE, and $\mathbf{w}^*$ is the optimal model we want to estimate. In Theorem 2, we show that

$$\|\mathbf{\Delta}_{t+1}\| \leq \max\{\Psi_t^{(1)}\|\mathbf{\Delta}_t\|, \Psi_t^{(2)}\|\mathbf{\Delta}_t\|^2\} + (\Psi_t^{(3)} + \alpha)\sqrt{\frac{1}{s}},$$

where $\{\Psi_t^{(i)}\}_{i=1}^3$ are quantities dependent on several problem parameters. Notice that the above implies a quadratic rate of convergence when $\|\mathbf{\Delta}_t\| \geq \Psi_t^{(1)}/\Psi_t^{(2)}$. Subsequently, when $\|\mathbf{\Delta}_t\|$ becomes sufficiently small, the above condition is violated and the convergence slows down to a linear rate. The error-floor, which is $\mathcal{O}(1/\sqrt{s})$ comes from the Byzantine resilience subroutine in conjunction with the simultaneous estimation of Hessian and gradient. Furthermore, in Section 5, we consider worker machines compressing the local Hessian inverse and gradient product via a ρ-approximate compressor [16], and show that the (order-wise) rate of convergence remain unchanged, and the compression factor, ρ affects the constants only.

We experimentally validate our proposed algorithm, COMRADE, with several benchmark data-sets. We consider several types of Byzantine attacks and observe that COMRADE is robust against

Byzantine worker machines, yielding better classification accuracy compared to the existing state-of-the-art second order algorithms.

A major technical challenge of this paper is to approximate local gradient and Hessian simultaneously in the presence of Byzantine workers. We use sketching, similar to [30], along with the norm based Byzantine resilience technique. Using *incoherence* (defined shortly) of the local Hessian along with concentration results originating from uniform sampling, we obtain the simultaneous gradient and Hessian approximation. Furthermore, ensuring at least one non-Byzantine machine gets trimmed at every iteration of COMRADE, we control the influence of Byzantine workers.

Related Work: *Second order Optimization:* Second order optimization has received a lot of attention in the recent years in the distributed setting owing to its attractive convergence speed. The fundamentals of second order optimization is laid out in [24], and an extension with better convergence rates is presented in [22]. Recently, in GIANT [30] algorithm, each worker machine computes an approximate Newton direction in each iteration and the center machine averages them to obtain a *globally improved* approximate Newton direction. Furthermore, DINGO [6] generalizes second order optimization beyond convex functions by extending the Newton-MR [23] algorithm in a distributed setting. Very recently, [8] proposes Determinantal averaging to correct the inversion bias of the second order optimization. A slightly different line of work ([29], [15], [21]) uses Hessian sketching to solve a large-scale distributed learning problems.

Byzantine Robust Optimization: In the seminal work of [12], a generic framework of one shot median based robust learning has been proposed and analyzed in the distributed setting. The issue of Byzantine failure is tackled by grouping the servers in batches and computing the median of batched servers in [5] (the median of means algorithm). Later in [32, 33], co-ordinate wise median, trimmed mean and iterative filtering based algorithm have been proposed and optimal statistical error rate is obtained. Also, [20, 7] consider adversaries may steer convergence to bad local minimizers for non-convex optimization problems.

Organization: In Section 3, we first analyze COMRADE with *one round* of communication per iteration. We assume $\alpha = 0$, and focus on the communication efficiency aspect only. Subsequently, in Section 4, we make $\alpha \neq 0$, thereby addressing communication efficiency and Byzantine resilience simultaneously. Further, in Section 5 we augment a compression scheme along with the setting of Section 4. Finally, in Section 6, we validate our theoretical findings with experiments. Proofs of all theoretical results can be found in the supplementary material.

Notation: For a positive integer r, $[r]$ denotes the set $\{1, 2, \ldots, r\}$. For a vector v, we use $\|v\|$ to denote the ℓ_2 norm unless otherwise specified. For a matrix X, we denote $\|X\|_2$ denotes the operator norm, $\sigma_{max}(X)$ and $\sigma_{min}(X)$ denote the maximum and minimum singular value. Throughout the paper, we use C, C_1, c, c_1 to denote positive universal constants, whose value changes with instances.

2 Problem Formulation

We begin with the standard statistical learning framework for empirical risk minimization, where the objective is to minimize the following loss function:

$$f(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} \ell_j(\mathbf{w}^T\mathbf{x}_j) + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{1}$$

where, the loss functions $\ell_j : \mathbb{R} \to \mathbb{R}$, $j \in [n]$ are *convex, twice differentiable and smooth*. Moreover, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ denote the input feature vectors and $y_1, y_2, \ldots, y_n \in \mathbb{R}$ denote the corresponding responses. Furthermore, we assume that the function f is *strongly convex*, implying the existence of a unique minimizer of (1). We denote this minimizer by $\mathbf{w}^*$. Note that the response $\{y_j\}_{j=1}^n$ is captured by the corresponding loss function $\{\ell_j\}_{j=1}^n$. Some examples of ℓ_j are

$$\text{logistic loss: } \ell_j(z_j) = \log(1 - \exp(-z_j y_j)), \quad \text{squared loss: } \ell_j(z_j) = \frac{1}{2}(z_j - y_j)^2$$

We consider the framework of distributed optimization with m worker machines, where the feature vectors and the loss functions $(\mathbf{x}_1, \ell_1), \ldots, (\mathbf{x}_n, \ell_n)$ are partitioned homogeneously among them. Furthermore, we assume that α fraction of the worker machines are Byzantine for some $\alpha < \frac{1}{2}$. The Byzantine machines, by nature, may send any arbitrary values to the center machine. Moreover, they can even collude with each other and plan malicious attacks with complete information of the learning algorithm.

3 COMRADE Can Communicate Less

We first present the Newton-type learning algorithm, namely COMRADE without any Byzantine workers, i.e., $\alpha = 0$. It is formally given in Algorithm 1 (with $\beta = 0$). In each iteration of our algorithm, every worker machine computes the local Hessian and local gradient and sends the local second order update (which is the product of the inverse of the local Hessian and local gradient) to the center machine. The center machine aggregates the updates from the worker machines by averaging them and updates the model parameter $\mathbf{w}$. Later the center machine broadcast the parameter $\mathbf{w}$ to all the worker machines.

In any iteration t, a standard Newton algorithm requires the computation of exact Hessian $(\mathbf{H}_t)$ and gradient $(\mathbf{g}_t)$ of the loss function which can be written as

$$\mathbf{g}_t = \frac{1}{n}\sum_{i=1}^{n} \ell_j'(\mathbf{w}_t^\top \mathbf{x}_i)\mathbf{x}_i + \lambda \mathbf{w}_t, \ \mathbf{H}_t = \frac{1}{n}\sum_{i=1}^{n} \ell_j''(\mathbf{w}_t^\top \mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^\top + \lambda \mathbf{I}. \tag{2}$$

In a distributed set up, the exact Hessian $(\mathbf{H}_t)$ and gradient $(\mathbf{g}_t)$ can be computed in parallel in the following manner. In each iteration, the center machine 'broadcasts' the model parameter $\mathbf{w}_t$ to the worker machines and each worker machine computes its own local gradient and Hessian. Then the center machine can compute the exact gradient and exact Hessian by averaging the the local gradient vectors and local Hessian matrices. But for each worker machine the per iteration communication complexity is $\mathcal{O}(d)$ for the gradient computation and $\mathcal{O}(d^2)$ for the Hessian computation. Using Algorithm 1, we reduce the communication cost to only $\mathcal{O}(d)$ per iteration, which is the same as the first order methods.

Each worker machine possess s samples drawn uniformly from $\{(\mathbf{x}_1, \ell_1), (\mathbf{x}_2, \ell_2), \ldots, (\mathbf{x}_n, \ell_n)\}$. By S_i, we denote the indices of the samples held by worker machine i. At any iteration t, the worker machine computes the local Hessian $\mathbf{H}_{i,t}$ and local gradient $\mathbf{g}_{i,t}$ as

$$\mathbf{g}_{i,t} = \frac{1}{s}\sum_{i\in S_i} \ell_j'(\mathbf{w}_t^\top \mathbf{x}_i)\mathbf{x}_i + \lambda \mathbf{w}_t, \quad \mathbf{H}_{i,t} = \frac{1}{s}\sum_{i\in S_i} \ell_j''(\mathbf{w}_t^\top \mathbf{x}_i)\mathbf{x}_i\mathbf{x}_i^\top + \lambda \mathbf{I}. \tag{3}$$

It is evident from the uniform sampling that $\mathbb{E}[\mathbf{g}_{i,t}] = \mathbf{g}_t$ and $\mathbb{E}[\mathbf{H}_{i,t}] = \mathbf{H}_t$. The update direction from the worker machine is defined as $\hat{\mathbf{p}}_{i,t} = (\mathbf{H}_{i,t})^{-1}\mathbf{g}_{i,t}$. Each worker machine requires $O(sd^2)$ operations to compute the Hessian matrix $\mathbf{H}_{i,t}$ and $O(d^3)$ operations to invert the matrix. In practice, the computational cost can be reduced by employing conjugate gradient method. The center machine computes the parameter update direction $\hat{\mathbf{p}}_t = \frac{1}{m}\sum_{i=1}^{m} \hat{\mathbf{p}}_{i,t}$.

We show that given large enough sample in each worker machine (s is large) and with incoherent data points (the information is spread out and not concentrated to a small number of sample data points), the local Hessian $\mathbf{H}_{i,t}$ is close to the global Hessian $\mathbf{H}_t$ in spectral norm, and the local gradient $\mathbf{g}_{i,t}$ is close to the global gradient $\mathbf{g}_t$. Subsequently, we prove that the empirical average of the local updates acts as a good proxy for the global Newton update and achieves good convergence guarantee.

3.1 Theoretical Guarantee

We define the matrix $\mathbf{A}_t^\top = [\mathbf{a}_1^\top, \ldots, \mathbf{a}_n^\top] \in \mathbb{R}^{d\times n}$ where $\mathbf{a}_j = \sqrt{\ell_j''(\mathbf{w}^\top \mathbf{x}_j)}\,\mathbf{x}_j$. So the exact Hessian in equation (2) is $\mathbf{H}_t = \frac{1}{n}\mathbf{A}_t^\top \mathbf{A}_t + \lambda \mathbf{I}$. Also we define $\mathbf{B}_t = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{d\times n}$ where $\mathbf{b}_i = \ell_i'(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i$. So the exact gradient in equation (2) is $\mathbf{g}_t = \frac{1}{n}\mathbf{B}_t\mathbf{1} + \lambda \mathbf{w}_t$

Definition 1 (Coherence of a Matrix)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be any matrix with $\mathbf{U} \in \mathbb{R}^{n\times d}$ being its orthonormal basis (the left singular vectors). The row coherence of the matrix $\mathbf{A}$ is defined as $\mu(\mathbf{A}) = \frac{n}{d}\max_i \|\mathbf{u}_i\|^2 \in \left[1, \frac{n}{d}\right]$, where $\mathbf{u}_i$ is the ith row of $\mathbf{U}$.*

Remark 1. *If the coherence of $\mathbf{A}_t$ is small, it can be shown that the Hessian matrix can be approximated well via selecting a subset of rows. Note that this is a fairly common to use coherence condition as an approximation tool (see [10, 11, 19])*

In the following, we assume that the Hessian matrix is L-Lipschitz (see definition below), which is a standard assumption for the analysis of the second order method for general smooth loss function (as seen in [30],[8]).

Assumption 1. *The Hessian matrix of the loss function f is L-Lipschitz continuous i.e. $\left\|\nabla^2 f(w) - \nabla^2 f(w')\right\|_2 \leq L\|w - w'\|$.*

Algorithm 1 COMmunication-efficient and Robust Approximate Distributed nEwton (COMRADE)

1: **Input:** Step size γ, parameter $\beta \geq 0$
2: **Initialize:** Initial iterate $w_0 \in \mathbb{R}^d$
3: **for** $t = 0, 1, \ldots, T-1$ **do**
4: Central machine: broadcasts w_t
for $i \in [m]$ **do in parallel**
5: i-th worker machine:
- Non-Byzantine: Computes local gradient $\mathbf{g}_{i,t}$ and local Hessian $\mathbf{H}_{i,t}$; sends $\hat{\mathbf{p}}_{i,t} = (\mathbf{H}_{i,t})^{-1}\mathbf{g}_{i,t}$ to the central machine,
- Byzantine: Generates $\star$ (arbitrary), and sends it to the center machine

end for
6: Center Machine:
- Sort the worker machines in a non decreasing order according to norm of updates $\{\hat{\mathbf{p}}_{i,t}\}_{i=1}^m$ from the local machines
- Return the indices of the first $1-\beta$ fraction of machines as $\mathcal{U}_t$,
- Approximate Newton Update direction : $\hat{\mathbf{p}}_t = \frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t}\hat{\mathbf{p}}_{i,t}$
- Update model parameter: $w_{t+1} = w_t - \gamma\hat{\mathbf{p}}_t$.

7: **end for**

In the following theorem, we provide the convergence rate of COMRADE (with $\alpha = \beta = 0$) in the terms of $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$. Also, we define $\kappa_t = \sigma_{max}(\mathbf{H}_t)/\sigma_{min}(\mathbf{H}_t)$ as the condition number of $\mathbf{H}_t$, and hence $\kappa_t \geq 1$.

Theorem 1. *Let $\mu \in \left[1, \frac{n}{d}\right]$ be the coherence of $\mathbf{A}_t$. Suppose $\gamma = 1$ and $s \geq \frac{3\mu d}{\eta^2}\log\frac{md}{\delta}$ for some $\eta, \delta \in (0,1)$. Under Assumption 1 , with probability exceeding $1-\delta$, we obtain*

$$\|\mathbf{\Delta}_{t+1}\| \leq \max\{\sqrt{\kappa_t(\frac{\zeta^2}{1-\zeta^2})}\|\mathbf{\Delta}_t\|, \frac{L}{\sigma_{min}(\mathbf{H}_t)}\|\mathbf{\Delta}_t\|^2\} + \frac{2\epsilon}{\sqrt{\sigma_{min}(\mathbf{H}_t)}},$$

where $\zeta = \nu(\frac{\eta}{\sqrt{m}} + \frac{\eta^2}{1-\eta})$, $\nu = \frac{\sigma_{max}(\mathbf{A}^\top\mathbf{A})}{\sigma_{max}(\mathbf{A}^\top\mathbf{A})+n\lambda} \leq 1$, and

$$\epsilon = \frac{1}{1-\eta}\frac{1}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}(1 + \sqrt{2\ln(\frac{m}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|. \tag{4}$$

Remark 2. *It is well known that a distributed Newton method has linear-quadratic convergence rate. In Theorem 1 the quadratic term comes from the standard analysis of Newton method. The linear term (which is small) arises owing to Hessian approximation. It gets smaller with better Hessian approximation (smaller η), and thus the above rate becomes quadratic one. The small error floor arises due to the gradient approximation in the worker machines, which is essential for the one round of communication per iteration. The error floor is $\propto \frac{1}{\sqrt{s}}$ where s is the number of samples in each worker machine. So for a sufficiently large s, the error floor becomes negligible.*

Remark 3. *The sample size in each worker machine is dependent on the coherence of the matrix $\mathbf{A}_t$ and the dimension d of the problem. Theoretically, the analysis is feasible for the case of $s \geq d$ (since we work with $\mathbf{H}_{i,t}^{-1}$). However, when $s < d$, one can replace the inverse by a pseudo-inverse (modulo some changes in convergence rate).*

4 COMRADE Can Resist Byzantine Workers

In this section, we analyze COMRADE with Byzantine workers. We assume that $\alpha(< 1/2)$ fraction of worker machines are Byzantine. We define the set of Byzantine worker machines by $\mathcal{B}$ and the set of the good (non-Byzantine) machines by $\mathcal{M}$. COMRADE employs a 'norm based thresholding' scheme on the local Hessian inverse and gradient product to tackle the Byzantine workers.

In the t-th iteration, the center machine outputs a set $\mathcal{U}_t$ with $|\mathcal{U}_t| = (1-\beta)m$, consisting the indices of the worker machines with smallest norm. Hence, we 'trim' the worker machines that may try to diverge the learning algorithm. We denote the set of trimmed machines as $\mathcal{T}_t$. Moreover, we take $\beta > \alpha$ to ensure at least one good machine falls in $\mathcal{T}_t$. This condition helps us to control the

Byzantine worker machines. Finally, the update is given by $\hat{\mathbf{p}}_t = \frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t}\hat{\mathbf{p}}_{i,t}$. We define:

$$\epsilon^2_{byz} = [3(\frac{1-\alpha}{1-\beta})^2 + 4\kappa_t(\frac{\alpha}{1-\beta})^2]\epsilon^2, \tag{5}$$

$$\zeta^2_{byz} = 2(\frac{1-\alpha}{1-\beta})^2(\frac{\nu}{1-\eta})^2 + \nu^2(\frac{1-\alpha}{1-\beta})^2(\frac{\eta}{\sqrt{(1-\alpha)m}} + \frac{\eta^2}{1-\eta})^2 + 4\kappa_t(\frac{\alpha}{1-\beta})^2[2 + (\frac{\nu}{1-\eta})^2]. \tag{6}$$

ϵ is defined in (4), $\nu = \frac{\sigma_{max}(\mathbf{A}^T\mathbf{A})}{\sigma_{max}(\mathbf{A}^T\mathbf{A})+n\lambda}$ and κ_t is the condition number of the exact Hessian $\mathbf{H}_t$.

Theorem 2. *Let $\mu \in \left[1, \frac{n}{d}\right]$ be the coherence of $\mathbf{A}_t$. Suppose $\gamma = 1$ and $s \geq \frac{3\mu d}{\eta^2}\log\frac{md}{\delta}$ for some $\eta, \delta \in (0,1)$. For $0 \leq \alpha < \beta < 1/2$, under Assumption 1 , with probability exceeding $1-\delta$, Algorithm 1 yields*

$$\|\mathbf{\Delta}_{t+1}\| \leq \max\{\sqrt{\kappa_t(\frac{\zeta^2_{byz}}{1-\zeta^2_{byz}})}\|\mathbf{\Delta}_t\|, \frac{L}{\sigma_{min}(\mathbf{H}_t)}\|\mathbf{\Delta}_t\|^2\} + \frac{2\epsilon_{byz}}{\sqrt{\sigma_{min}(\mathbf{H}_t)}},$$

where ζ_{byz} and ϵ_{byz} are defined in equations (5) *and* (6) *respectively.*

The remarks of Section 3 is also applicable here. On top of that, we have the following remarks:

Remark 4. *Compared to the convergence rate of Theorem 1, the rate here remains order-wise same even with Byzantine robustness. The coefficient of the quadratic term remains unchanged but the linear rate and the error floor suffers a little bit (by a small constant factor).*

Remark 5. *Note that for Theorem 2 to hold, we require $\alpha \sim 1/\sqrt{\kappa_t}$ for all t. In cases where κ_t is large, this can impose a stricter condition on α. However, we conjecture that this dependence can be improved via applying a more intricate (and perhaps computation heavy) Byzantine resilience algorithm. In this work, we kept the Byzantine resilience scheme simple at the expense of this condition on α.*

5 COMRADE Can Communicate Even Less and Resist Byzantine Workers

In Section 3 we analyze COMRADE with an additional feature. We let the worker machines further reduce the communication cost by applying a generic class of ρ-approximate compressor [16] on the parameter update of Algorithm 1. We first define the class of ρ-approximate compressor:

Definition 2. *An operator $\mathcal{Q}: \mathbb{R}^d \to \mathbb{R}^d$ is defined as ρ-approximate compressor on a set $S \subset \mathbb{R}^d$ if, $\forall x \in S$, $\|\mathcal{Q}(x) - x\|^2 \leq (1-\rho)\|x\|^2$, where $\rho \in [0,1]$ is the compression factor.*

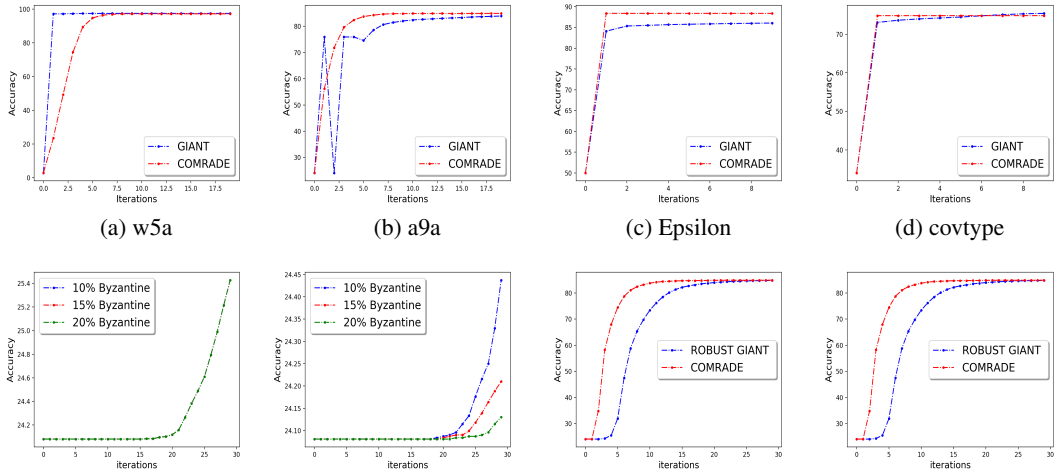
The above definition can be extended for any randomized operator $\mathcal{Q}$ satisfying $\mathbb{E}(\|\mathcal{Q}(x) - x\|^2) \leq (1-\rho)\|x\|^2$, for all $\forall x \in S$. The expectation is taken over the randomization of the operator. Notice that $\rho = 1$ implies that $\mathcal{Q}(x) = x$ (no compression). Examples of ρ-approximate compressor include QSGD [1], ℓ_1-QSGD [16], top$_k$ sparsification and rand$_k$ [25].

Worker machine i computes the product of local Hessian inverse inverse and local gradient and then apply ρ-approximate compressor to obtain $\mathcal{Q}(\mathbf{H}^{-1}_{i,t}\mathbf{g}_{i,t})$; and finally sends this compressed vector to the center. The Byzantine resilience subroutine remains the same–except, instead of sorting with respect to $\|\mathbf{H}^{-1}_{i,t}\mathbf{g}_{i,t}\|$, the center machine now sorts according to $\|\mathcal{Q}(\mathbf{H}^{-1}_{i,t}\mathbf{g}_{i,t})\|$. The center machine aggregates the compressed updates by averaging $\mathcal{Q}(\hat{\mathbf{p}}) = \frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t}\mathcal{Q}(\hat{\mathbf{p}}_{i,t})$, and take the next step as $w_{t+1} = w_t - \gamma\mathcal{Q}(\hat{\mathbf{p}})$.

Recall the definition of ϵ from (4). We also use the following notation : $\zeta^2_{\mathcal{M}} = \nu(\frac{\eta}{\sqrt{(1-\alpha)m}} + \frac{\eta^2}{1-\eta})$, $\zeta_1 = \frac{\nu}{1-\eta}$ and $\nu = \frac{\sigma_{max}(\mathbf{A}^T\mathbf{A})}{\sigma_{max}(\mathbf{A}^T\mathbf{A})+n\lambda}$. Furthermore, we define the following:

$$\epsilon^2_{comp,byz} = [3(\frac{1-\alpha}{1-\beta})^2 + 4\kappa_t(\frac{\alpha}{1-\beta})^2](1+\kappa(1-\rho))\epsilon^2 \tag{7}$$

$$\begin{aligned}\zeta^2_{comp,byz} = {} & 2(\frac{1-\alpha}{1-\beta})^2(\zeta_1^2 + \kappa_t(1-\rho)((1+\zeta_1^2)) + (\frac{1-\alpha}{1-\beta})^2(\zeta^2_{\mathcal{M}} + \kappa_t(1-\rho)((1+\zeta_1^2)) \\ & + 4\kappa_t(\frac{\alpha}{1-\beta})^2(2 + (\zeta_1^2 + \kappa_t(1-\rho)((1+\zeta_1^2)))\end{aligned} \tag{8}$$



(a) w5a (b) a9a (c) Epsilon (d) covtype

(e) GIANT 'flipped' attack (f) GIANT 'negative' attack (g) Robust GIANT (h) Robust GIANT

Figure 1: (First row) Comparison of training accuracy between COMRADE(Algorithm 1) and GIANT [30] with (a) w5a (b) a9a (c) Epsilon (d) Covtype dataset. (Second row) Training accuracy of (e) GIANT for 'flipped label' and (f) 'negative update' attack; and comparison of Robust GIANT and COMRADE with a9a dataset for (g) 'flipped label' and (h) 'negative update' attack.

Theorem 3. *Let $\mu \in \left[1, \frac{n}{d}\right]$ be the coherence of $\mathbf{A}_t$. Let $\gamma = 1$ and $s \geq \frac{3\mu d}{\eta^2} \log \frac{md}{\delta}$ for some $\eta, \delta \in (0,1)$. For $0 \leq \alpha < \beta < 1/2$, under Assumption 1 and with $\mathcal{Q}$ being the ρ-approximate compressor, with probability exceeding $1-\delta$, we obtain*

$$\|\mathbf{\Delta}_{t+1}\| \leq \max\{\sqrt{\kappa_t(\frac{\zeta^2_{comp,byz}}{1-\zeta^2_{comp,byz}})}\|\mathbf{\Delta}_t\|, \frac{L}{\sigma_{min}(\mathbf{H}_t)}\|\mathbf{\Delta}_t\|^2\} + \frac{\epsilon_{comp,byz}}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}$$

where $\epsilon_{comp,byz}$ and $\zeta_{comp,byz}$ are given in equations (7) *and* (8) *respectively.*

Remark 6. *With no compression ($\rho = 1$) we get back the convergence guarantee of Theorem 2.*

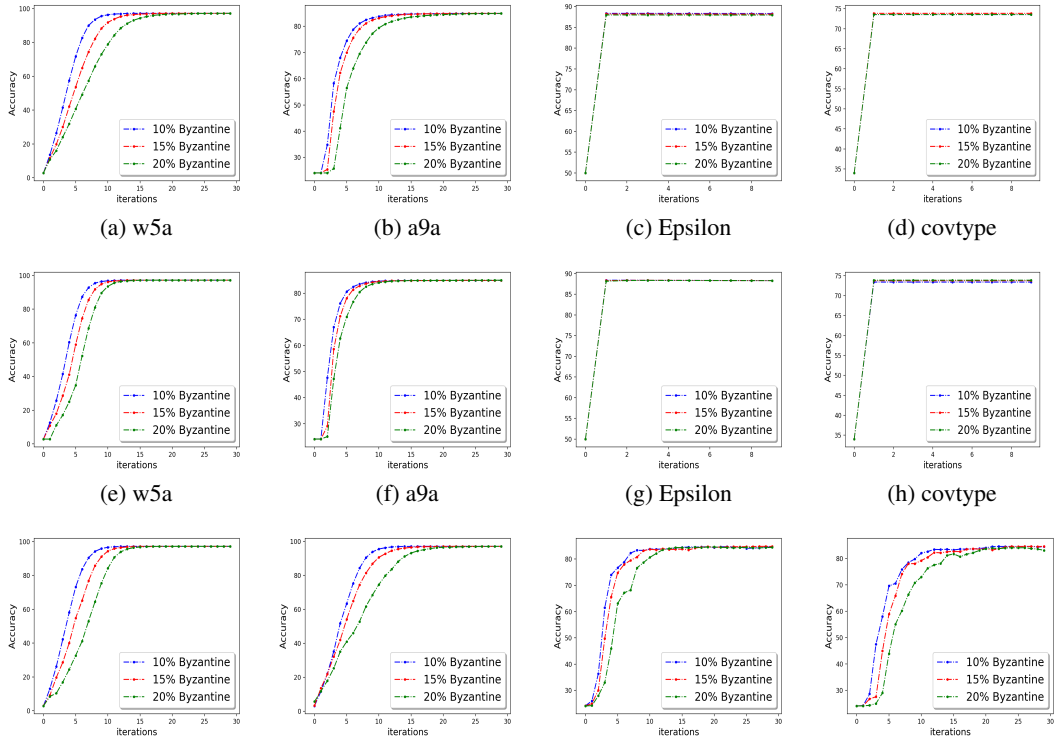
Remark 7. *Note that even with compression, we retain the linear-quadratic rate of convergence of COMRADE. The constants are affected by a ρ-dependent term.*

6 Experimental Results

In this section we validate our algorithm, COMRADE in Byzantine and non-Byzantine setup on synthetically generated and benchmark LIBSVM [4] data-set. The experiments focus on the standard logistic regression problem. The logistic regression objective is defined as $\frac{1}{n}\sum_{i=1}^{n} \log\left(1+\exp(-y_i \mathbf{x}_i^\top \mathbf{w})\right) + \frac{\lambda}{2n}\|\mathbf{w}\|^2$, where $\mathbf{w} \in \mathbb{R}^d$ is the parameter, $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^d$ are the feature data and $\{y_i\}_{i=1}^n \in \{0,1\}$ are the corresponding labels. We use 'mpi4py' package for distributed framework in a computing cluster[1]. We choose 'a9a' ($d = 123, n \approx 32$K), 'w5a' ($d = 300, n \approx 10k$), 'Epsilon' ($d = 2000, n = 0.4$M) and 'covtype.binary' ($d = 54, n \approx 0.5$M) classification datasets and partition the data in 20 different worker machines. In the experiments, we choose two types of Byzantine attacks : (1). 'flipped label'-attack where (for binary classification) the Byzantine worker machines flip the labels of the data, thus making the model learn with wrong labels, and (2). 'negative update attack' where the Byzantine worker machines compute the local update ($\hat{\mathbf{p}}_i$) and communicate $-c \times \hat{\mathbf{p}}_i$ with $c \in (0,1)$ making the updates to be opposite of actual direction. We choose $\beta = \alpha + \frac{2}{m}$.

In Figure 1(first row) we compare COMRADE in non-Byzantine setup ($\alpha = \beta = 0$) with the state-of the art algorithm GIANT [30]. It is evident from the plot that despite the fact that COMRADE requires less communication, the algorithm is able to achieve similar accuracy. Also, we show the

[1]The cluster information is absent for anonymity.



(a) w5a (b) a9a (c) Epsilon (d) covtype

(e) w5a (f) a9a (g) Epsilon (h) covtype

(i) w5a 'flipped' (j) w5q 'negative' (k) a9a 'flipped' (l) a9a 'negative'

Figure 2: (First row) Accuracy of COMRADE with $10\%, 15\%, 20\%$ Byzantine workers with 'negative update' attack for (a). w5a (b). a9a (c). covtype (d). Epsilon. (Second row) COMRADE accuracy with $10\%, 15\%, 20\%$ Byzantine workers with 'flipped label' attack for (e) w5a (f) a9a (g) covtype (h) Epsilon. (Third row) Accuracy of COMRADE with ρ-approximate compressor (Section 5) with $10\%, 15\%, 20\%$ Byzantine workers; (i) 'flipped label' attack for w5a (j) 'negative update' attack for w5a. (k) 'flipped label' attack for a9a . (l) 'negative update' attack for a9a dataset.

ineffectiveness of GIANT in the presence of Byzantine attacks. In Figure 2((e),(f)) we show the accuracy for flipped label and negative update attacks. These plots are an indicator of the requirement of robustness in the learning algorithm. So we device 'Robust GIANT', which is GIANT algorithm with added 'norm based thresholding' for robustness. In particular, we trim the worker machines based on the local gradient norm in the first round of communication of GIANT. Subsequently, in the second round of communication, the non-trimmed worker machines send the updates (product of local Hessian inverse and the local gradient) to the center machine. We compare COMRADE with 'Robust GIANT' in Figure 1((g),(h)) with 10% Byzantine worker machines for 'a9a' dataset. It is evident plot that COMRADE performs better than the 'Robust GIANT'.

Next we show the accuracy of COMRADE with different numbers of Byzantine worker machines. Here we choose $c = 0.9$. We show the accuracy for 'negetive update ' attack in Figure 2(first row) and 'flipped label' attack in Figure 2 (second row). Furthermore, we show that COMRADE works even when ρ-approximate compressor is applied to the updates. In Figure 2(Third row) we plot the tranning accuracies. For compression we apply the scheme known as QSGD [1]. Further experiments can be found in the supplementary material.

7 Conclusion and Future Work

In this paper, we address the issue of communication efficiency and Byzantine robustness via second order optimization and norm based thresholding respectively for strongly convex loss. Extending our setting to handle weakly convex and non-convex loss is of immediate interest. We would also like to exploit local averaging with second order optimization. Moreover, an import aspect, privacy, is not addressed in this work. We keep this as our future research direction.

Broader Impact

The advent of computationally-intensive machine learning (ML) models has changed the technology landscape in the past decade. The most powerful learning models are also the most expensive to train. For example, OpenAI's GPT-3 language model has 175 billion parameters and takes USD 12 million to train[2]! On top of that machine learning training has a costly environmental footprint: recent study shows that training a transformer with neural architecture search can have as much as five times CO_2 emission of a standard car in its lifetime[3]. While the really expensive models are relatively rare, training of moderately large ML models is now ubiquitous over the data science industry and elsewhere. Most of the training of machine learning model today is performed in distributed platforms (such as Amazon's EC2). Any savings in energy - in forms of computation or communication - in distributed optimization will have a large positive impact.

This paper seeks to speed up distributed optimization algorithms by minimizing inter-server communication and at the same time makes the optimization algorithms robust to adversarial failures. The protocols resulting from this paper are immediately implementable and can be adapted to any large scale distributed training of a machine learning model. Further, since our algorithms are robust to Byzantine failure, the training process becomes more reliable and fail-safe.

In addition to that, we think the theoretical content of this paper is instructive and some elements can be included in the coursework of a graduate class of distributed optimization, to exemplify the trade-off between some fundamental quantities in distributed optimization.

References

[1] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signsgd: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

[3] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer. Byzantine-tolerant machine learning. *arXiv preprint arXiv:1703.02757*, 2017.

[4] 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.

[5] Y. Chen, L. Su, and J. Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

[6] R. Crane and F. Roosta. Dingo: Distributed newton-type method for gradient-norm optimization. In *Advances in Neural Information Processing Systems*, pages 9494–9504, 2019.

[7] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, A. H. A. Guirguis, and S. L. A. Rouault. Aggregathor: Byzantine machine learning via robust gradient aggregation. page 19, 2019. Published in the Conference on Systems and Machine Learning (SysML) 2019, Stanford, CA, USA.

[8] M. Derezinski and M. W. Mahoney. Distributed estimation of the inverse hessian by determinantal averaging. In *Advances in Neural Information Processing Systems*, pages 11401–11411, 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(Dec):3475–3506, 2012.

[11] P. Drineas and M. W. Mahoney. Randnla: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[2] `https://venturebeat.com/2020/06/01/ai-machine-learning-openai-gpt-3-size-isnt-everything/`

[3] MIT Tech. Review article dated 2019/06/06

[12] J. Feng, H. Xu, and S. Mannor. Distributed robust learning. *arXiv preprint arXiv:1409.5937*, 2014.

[13] V. Gandikota, R. K. Maity, and A. Mazumdar. vqsgd: Vector quantized stochastic gradient descent. *arXiv preprint arXiv:1911.07971*, 2019.

[14] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[15] V. Gupta, S. Kadhe, T. Courtade, M. W. Mahoney, and K. Ramchandran. Oversketched newton: Fast convex optimization for serverless systems. *arXiv preprint arXiv:1903.08857*, 2019.

[16] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi. Error feedback fixes signsgd and other gradient compression schemes. *arXiv preprint arXiv:1901.09847*, 2019.

[17] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[18] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[19] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

[20] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault. The hidden vulnerability of distributed learning in byzantium. *arXiv preprint arXiv:1802.07927*, 2018.

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

[22] S. J. Reddi, J. Konečnỳ, P. Richtárik, B. Póczós, and A. Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

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

[24] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008, 2014.

[25] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.

[26] L. Su and N. H. Vaidya. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM symposium on principles of distributed computing*, pages 425–434. ACM, 2016.

[27] A. T. Suresh, F. X. Yu, S. Kumar, and H. B. McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3329–3337. JMLR. org, 2017.

[28] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018.

[29] S. Wang, A. Gittens, and M. W. Mahoney. Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging. *The Journal of Machine Learning Research*, 18(1):8039–8088, 2017.

[30] S. Wang, F. Roosta, P. Xu, and M. W. Mahoney. Giant: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, pages 2332–2342, 2018.

[31] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems*, pages 1509–1519, 2017.

[32] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659, StockholmsmÃ€ssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[33] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Defending against saddle point attack in Byzantine-robust distributed learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7074–7084, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[34] Y. Zhang and X. Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370, 2015.

Distributed Newton Can Communicate Less and Resist Byzantine Workers
Supplementary Material

8 Appendix A: Analysis of Section 3

Matrix Sketching
Here we briefly discuss the matrix sketching that is broadly used in the context of *randomized linear algebra*. For any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ the sketched matrix $\mathbf{Z} \in \mathbb{R}^{s\times d}$ is defined as $\mathbf{S}^T\mathbf{A}$ where $\mathbf{S} \in \mathbb{R}^{n\times s}$ is the sketching matrix (typically $s < n$). Based on the scope and basis of the application, the sketched matrix is constructed by taking linear combination of the rows of matrix which is known as *random projection* or by sampling and scaling a subset of the rows of the matrix which is known as *random sampling*. The sketching is done to get a smaller representation of the original matrix to reduce computational cost.

Here we consider a uniform row sampling scheme. The matrix $\mathbf{Z}$ is formed by sampling and scaling rows of the matrix $\mathbf{A}$. Each row of the matrix $\mathbf{A}$ is sampled with probability $p = \frac{1}{n}$ and scaled by multiplying with $\frac{1}{\sqrt{sp}}$.

$$\mathbb{P}\left(\mathbf{z}_i = \frac{\mathbf{a}_j}{\sqrt{sp}}\right) = p,$$

where $\mathbf{z}_i$ is the i-th row matrix $\mathbf{Z}$ and $\mathbf{a}_j$ is the j th row of the matrix $\mathbf{A}$. Consequently the sketching matrix $\mathbf{S}$ has one non-zero entry in each column.

We define the matrix $\mathbf{A}_t^\top = [\mathbf{a}_1^\top, \ldots, \mathbf{a}_n^\top] \in \mathbb{R}^{d\times n}$ where $\mathbf{a}_j = \sqrt{\ell_j''(\mathbf{w}^\top\mathbf{x}_j)}\,\mathbf{x}_j$. So the exact Hessian in equation (2) is $\mathbf{H}_t = \frac{1}{n}\mathbf{A}_t^\top\mathbf{A}_t + \lambda\mathbf{I}$. Assume that S_i is the set of features that are held by the ith worker machine. So the local Hessian is

$$\mathbf{H}_{i,t} = \frac{1}{s}\sum_{j\in S_i} \ell_j''(\mathbf{w}^\top\mathbf{x}_j)\mathbf{x}_j\mathbf{x}_j^\top + \lambda\mathbf{I} = \frac{1}{s}\mathbf{A}_{i,t}^\top\mathbf{A}_{i,t} + \lambda\mathbf{I},$$

where $\mathbf{A}_{i,t} \in \mathbb{R}^{s\times d}$ and the row of the matrix $\mathbf{A}_{i,t}$ is indexed by S_i. Also we define $\mathbf{B}_t = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{d\times n}$ where $\mathbf{b}_i = \ell_i'(\mathbf{w}^\top\mathbf{x}_i)\mathbf{x}_i$. So the exact gradient in equation (2) is $\mathbf{g}_t = \frac{1}{n}\mathbf{B}_t\mathbf{1} + \lambda\mathbf{w}_t$ and the local gradient is

$$\mathbf{g}_{i,t} = \frac{1}{s}\sum_{i\in S_i} \ell_j'(\mathbf{w}_t^\top\mathbf{x}_i)\mathbf{x}_i + \lambda\mathbf{w}_t = \frac{1}{s}\mathbf{B}_{i,t}\mathbf{1} + \lambda\mathbf{w}_t,$$

where $\mathbf{B}_{i,t}$ is the matrix with column indexed by S_i. If $\{\mathbf{S}_i\}_{i=1}^m$ are the sketching matrices then the local Hessian and gradient can be expressed as

$$\mathbf{H}_{i,t} = \mathbf{A}_t^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{A}_t^\top + \lambda\mathbf{I} \qquad\qquad \mathbf{g}_{i,t} = \frac{1}{n}\mathbf{B}\mathbf{S}_i\mathbf{S}_i^\top\mathbf{1} + \lambda\mathbf{w}. \tag{9}$$

With the help of sketching idea later we show that the local hessian and gradient are close to the exact hessian and gradient.

The Quadratic function For the purpose of analysis we define an auxiliary quadratic function

$$\phi(\mathbf{p}) = \frac{1}{2}\mathbf{p}^\top\mathbf{H}_t\mathbf{p} - \mathbf{g}_t^\top\mathbf{p} = \frac{1}{2}\mathbf{p}^\top(\mathbf{A}_t^\top\mathbf{A}_t + \lambda\mathbf{I})\mathbf{p} - \mathbf{g}_t^\top\mathbf{p}. \tag{10}$$

The optimal solution to the above function is

$$\mathbf{p}^* = \arg\min\phi(\mathbf{p}) = \mathbf{H}_t^{-1}\mathbf{g}_t = (\mathbf{A}_t^\top\mathbf{A}_t + \lambda\mathbf{I})^{-1}\mathbf{g}_t,$$

which is also the optimal direction of the global Newton update. In this work we consider the local and global (approximate) Newton direction to be

$$\hat{\mathbf{p}}_{i,t} = (\mathbf{A}^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{g}_{i,t}, \quad \hat{\mathbf{p}}_t = \frac{1}{m}\sum_{i=1}^m \hat{\mathbf{p}}_{i,t}.$$

respectively. And it can be easily verified that each local update $\hat{\mathbf{p}}_{i,t}$ is optimal solution to the following quadratic function

$$\hat{\phi}_{i,t}(p) = \frac{1}{2}\mathbf{p}^\top(\mathbf{A}^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{A} + \lambda\mathbf{I})\mathbf{p} - \mathbf{g}_i^\top\mathbf{p}. \tag{11}$$

In our convergence analysis we show that value of the quadratic function in (10) with value $\hat{\mathbf{p}}_t$ is close to the optimal value.

Singular Value Decomposition (SVD) For any matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rank r, the singular value decomposition is defined as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ where $\mathbf{U}, \mathbf{V}$ are $n \times r$ and $d \times r$ column orthogonal matrices respectively and $\mathbf{\Sigma}$ is a $r \times r$ diagonal matrix with diagonal entries $\{\sigma_1, \dots \sigma_r\}$. If $\mathbf{A}$ is a symmetric positive semi-definite matrix then $\mathbf{U} = \mathbf{V}$.

8.1 Analysis

Lemma 1 (McDiarmid's Inequality)**.** *Let $X = X_1, \dots, X_m$ be m independent random variables taking values from some set A, and assume that $f : A^m \to \mathbb{R}$ satisfies the following condition (bounded differences):*

$$\sup_{x_1,\dots,x_m,\hat{x}_i} |f(x_i, \dots, x_i, \dots, x_m) - f(x_i, \dots, \hat{x}_i, \dots, x_m)| \le c_i,$$

for all $i \in \{1, \dots, m\}$. Then for any $\epsilon > 0$ we have

$$P\left[f(X_1, \dots, X_m) - \mathbb{E}[f(X_1, \dots, X_m)] \ge \epsilon\right] \le \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^m c_i^2}\right).$$

The property described in the following Lemma 2 is a very useful result for uniform row sampling sketching matrix.

Lemma 2 (Lemma 8 [30])**.** *Let $\eta, \delta \in (0, 1)$ be a fixed parameter and $r = rank(\mathbf{A}_t)$ and $\mathbf{U} \in \mathbb{R}^{n\times r}$ be the orthonormal bases of the matrix $\mathbf{A}_t$. Let $\{\mathbf{S}_i\}_{i=1}^m$ be sketching matrices and $\mathbf{S} = \frac{1}{\sqrt{m}}[\mathbf{S}_1, \dots \mathbf{S}_m] \in \mathbb{R}^{n\times ms}$. With probability $1 - \delta$ the following holds*

$$\left\|\mathbf{U}^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{U} - \mathbf{I}\right\|_2 \le \eta \quad \forall i \in [m] \quad \textit{and} \quad \left\|\mathbf{U}^\top\mathbf{S}\mathbf{S}^\top\mathbf{U} - \mathbf{I}\right\|_2 \le \frac{\eta}{\sqrt{m}}.$$

Lemma 3. *Let $\mathbf{S} \in \mathbb{R}^{n\times s}$ be any uniform sampling sketching matrix, then for any matrix $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] \in \mathbb{R}^{d\times n}$ with probability $1 - \delta$ for any $\delta > 0$ we have,*

$$\|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1} - \frac{1}{n}\mathbf{B}\mathbf{1}\| \le (1 + \sqrt{2\ln(\frac{1}{\delta})})\sqrt{\frac{1}{s}}\max_i \|\mathbf{b}_i\|,$$

where $\mathbf{1}$ is all ones vector.

Proof. The vector $\mathbf{B}\mathbf{1}$ is the sum of column of the matrix $\mathbf{B}$ and $\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1}$ is the sum of uniformly sampled and scaled column of the matrix $\mathbf{B}$ where the scaling factor is $\frac{1}{\sqrt{sp}}$ with $p = \frac{1}{n}$. If $(i_1, \dots, i_s)$ is the set of sampled indices then $\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1} = \sum_{k\in(i_1,\dots,i_s)} \frac{1}{sp}\mathbf{b}_k$.

Define the function $f(i_1, \dots, i_s) = \|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1} - \frac{1}{n}\mathbf{B}\mathbf{1}\|$. Now consider a sampled set $(i_1, \dots, i_{j'}, \dots, i_s)$ with only one item (column) replaced then the bounded difference is

$$\begin{aligned}\Delta &= |f(i_1, \dots, i_j, \dots, i_s) - f(i_1, \dots, i_{j'}, \dots, i_s)| \\ &= |\frac{1}{n}\|\frac{1}{sp}\mathbf{b}_{i'_j} - \frac{1}{sp}\mathbf{b}_{i_j}\|| \le \frac{2}{s}\max_i \|\mathbf{b}_i\|.\end{aligned}$$

Now we have the expectation

$$\begin{aligned}&\mathbb{E}[\|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1} - \frac{1}{n}\mathbf{B}\mathbf{1}\|^2] \le \frac{n}{sn^2}\sum_{i=1}^n \|\mathbf{b}_i\|^2 = \frac{1}{s}\max_i \|\mathbf{b}_i\|^2 \\ \Rightarrow\ &\mathbb{E}[\|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1} - \frac{1}{n}\mathbf{B}\mathbf{1}\|] \le \sqrt{\frac{1}{s}}\max_i \|\mathbf{b}_i\|.\end{aligned}$$

Using McDiarmid inequality (Lemma 1) we have

$$P[\left\|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1}-\frac{1}{n}\mathbf{B}\mathbf{1}\right\|\geq\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|+t\Bigg]\leq\exp\left(-\frac{2t^2}{s\Delta^2}\right).$$

Equating the probability with δ we have

$$\begin{aligned}&\exp(-\frac{2t^2}{s\Delta^2})=\delta\\ \Rightarrow&t=\Delta\sqrt{\frac{s}{2}\ln(\frac{1}{\delta})}=\max_i\|\mathbf{b}_i\|\sqrt{\frac{2}{s}\ln(\frac{1}{\delta})}.\end{aligned}$$

Finally we have with probability $1-\delta$

$$\|\frac{1}{n}\mathbf{B}\mathbf{S}\mathbf{S}^\top\mathbf{1}-\frac{1}{n}\mathbf{B}\mathbf{1}\|\leq(1+\sqrt{2\ln(\frac{1}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|.$$

□

Remark 8. *For m sketching matrix $\{\mathbf{S}_i\}_{i=1}^m$, the bound in the Lemma 3 is*

$$\|\frac{1}{n}\mathbf{B}\mathbf{S}_i\mathbf{S}_i^\top\mathbf{1}-\frac{1}{n}\mathbf{B}\mathbf{1}\|\leq(1+\sqrt{2\ln(\frac{m}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|,$$

with probability $1-\delta$ for any $\delta>0$ for all $i\in\{1,2,\ldots,m\}$. In the case that each worker machine holds data based on the uniform sketching matrix the local gradient is close to the exact gradient. Thus the local second order update acts as a good approximate to the exact Netwon update.

Now we consider the update rule of GIANT [30] where the update is done in two rounds in each iteration. In the first round each worker machine computes and send the local gradient and the center machine computes the exact gradient $\mathbf{g}_t$ in iteration t. Next the center machine broadcasts the exact gradient and each worker machine computes the local Hessian and send $\tilde{\mathbf{p}}_{i,t}=(\mathbf{H}_{i,t})^{-1}\mathbf{g}_t$ to the center machine and the center machine computes the approximate Newton direction $\tilde{\mathbf{p}}_t=\frac{1}{m}\sum_{i=1}^m\tilde{\mathbf{p}}_{i,t}$. Now based on this we restate the following lemma (Lemma 6 [30]).

Lemma 4. *Let $\{\mathbf{S}_i\}_{i=1}^m\in\mathbb{R}^{n\times s}$ be sketching matrices based on Lemma 2. Let ϕ_t be defined in* (10) *and $\tilde{\mathbf{p}}_t$ be the update. It holds that*

$$\min_{\mathbf{p}}\phi_t(\mathbf{p})\leq\phi_t(\tilde{\mathbf{p}}_t)\leq(1-\zeta^2)\min_{\mathbf{p}}\phi_t(\mathbf{p}),$$

where $\zeta=\nu(\frac{\eta}{\sqrt{m}}+\frac{\eta^2}{1-\eta})$ and $\nu=\frac{\sigma_{max}(\mathbf{A}^\top\mathbf{A})}{\sigma_{max}(\mathbf{A}^\top\mathbf{A})+n\lambda}\leq 1$.

Now we prove similar guarantee for the update according to COMRADE in Algorithm 1.

Lemma 5. *Let $\{\mathbf{S}_i\}_{i=1}^m\in\mathbb{R}^{n\times s}$ be sketching matrices based on Lemma 2. Let ϕ_t be defined in* (10) *and $\hat{\mathbf{p}}_t$ be defined in Algorithm 1($\beta=0$)*

$$\min_{\mathbf{p}}\phi_t(\mathbf{p})\leq\phi_t(\hat{\mathbf{p}}_t)\leq\epsilon^2+(1-\zeta^2)\min_{\mathbf{p}}\phi_t(\mathbf{p}),$$

where $\epsilon=\frac{1}{1-\eta}\frac{1}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}(1+\sqrt{2\ln(\frac{m}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|$ and $\zeta=\nu(\frac{\eta}{\sqrt{m}}+\frac{\eta^2}{1-\eta})$ and $\nu=\frac{\sigma_{max}(\mathbf{A}^\top\mathbf{A})}{\sigma_{max}(\mathbf{A}^\top\mathbf{A})+n\lambda}$.

Proof. First consider the quadratic function (10)

$$\begin{aligned}\phi_t(\hat{\mathbf{p}}_t)-\phi_t(\mathbf{p}^*)&=\frac{1}{2}\|\mathbf{H}_t^{\frac{1}{2}}(\hat{\mathbf{p}}_t-\mathbf{p}^*)\|^2\\ &\leq\underbrace{(\|\mathbf{H}_t^{\frac{1}{2}}(\hat{\mathbf{p}}_t-\tilde{\mathbf{p}}_t)\|^2}_{Term1}+\underbrace{\|\mathbf{H}_t^{\frac{1}{2}}(\tilde{\mathbf{p}}_t-\mathbf{p}^*)\|)^2}_{Term2},\end{aligned}\tag{12}$$

where $\tilde{\mathbf{p}}_t = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{H}_{i,t})^{-1}\mathbf{g}_t$. First we bound the Term 2 of (12) using the quadratic function and Lemma 4

$$\begin{aligned}\frac{1}{2}\left\|\mathbf{H}_t^{\frac{1}{2}}(\tilde{\mathbf{p}}_t - \mathbf{p}^*)\right\|)^2 &\leq \zeta^2\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{p}^*\right\|^2 \quad \text{(Using Lemma 4)}\\ &= -\zeta^2\phi_t(\mathbf{p}^*). \end{aligned} \tag{13}$$

The step in equation (13) is from the definition of the function ϕ_t and $\mathbf{p}^*$. It can be shown that

$$\phi_t(\mathbf{p}^*) = -\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{p}^*\right\|^2.$$

Now we bound the Term 1 in (12). By Lemma 2, we have $(1-\eta)\mathbf{A}_t^\top\mathbf{A}_t \preceq \mathbf{A}_t^\top\mathbf{S}_i\mathbf{S}_i^\top\mathbf{A}_t \preceq (1+\eta)\mathbf{A}_t^\top\mathbf{A}_t$. Following we have $(1-\eta)\mathbf{H}_t \preceq \mathbf{H}_{i,t} \preceq (1+\eta)\mathbf{H}_t$. Thus there exists matrix ξ_i satisfying

$$\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_{i,t}^{-1}\mathbf{H}_t^{\frac{1}{2}} = \mathbf{I} + \xi_i \quad \text{and} \quad -\frac{\eta}{1+\eta} \preceq \xi_i \preceq \frac{\eta}{1-\eta},$$

So we have,

$$\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_{i,t}^{-1}\mathbf{H}_t^{\frac{1}{2}}\right\| \leq 1 + \frac{\eta}{1-\eta} = \frac{1}{1-\eta}. \tag{14}$$

Now we have

$$\begin{aligned}\left\|\mathbf{H}_t^{\frac{1}{2}}(\hat{\mathbf{p}}_t - \tilde{\mathbf{p}}_t)\right\| &= \left\|\mathbf{H}_t^{\frac{1}{2}}\frac{1}{m}\sum_{i=1}^{m}(\hat{\mathbf{p}}_{i,t} - \tilde{\mathbf{p}}_{i,t})\right\|\\ &\leq \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_t^{\frac{1}{2}}(\hat{\mathbf{p}}_{i,t} - \tilde{\mathbf{p}}_{i,t})\right\|\\ &= \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_{i,t}^{-1}(\mathbf{g}_{i,t} - \mathbf{g}_t)\right\|\\ &= \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_{i,t}^{-1}\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_t^{-\frac{1}{2}}(\mathbf{g}_{i,t} - \mathbf{g}_t)\right\|\\ &\leq \frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_t^{\frac{1}{2}}\mathbf{H}_{i,t}^{-1}\mathbf{H}_t^{\frac{1}{2}}\right\|\left\|\mathbf{H}_t^{-\frac{1}{2}}(\mathbf{g}_{i,t} - \mathbf{g}_t)\right\|\\ &\leq \frac{1}{1-\eta}\frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{H}_t^{-\frac{1}{2}}(\mathbf{g}_{i,t} - \mathbf{g}_t)\right\| \quad \text{(Using (14))}\\ &\leq \frac{1}{1-\eta}\frac{1}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}\frac{1}{m}\sum_{i=1}^{m}\left\|(\mathbf{g}_{i,t} - \mathbf{g}_t)\right\|. \end{aligned} \tag{15}$$

Now we bound $\|(\mathbf{g}_{i,t} - \mathbf{g}_t)\|$ using Lemma 3,

$$\|(\mathbf{g}_{i,t} - \mathbf{g}_t)\| = \|\frac{1}{n}\mathbf{BSS}^\top\mathbf{1} - \frac{1}{n}\mathbf{B1}\| \leq (1 + \sqrt{2\ln(\frac{m}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|.$$

Plugging it into equation (15) we get,

$$\begin{aligned}\left\|\mathbf{H}_t^{\frac{1}{2}}(\hat{\mathbf{p}}_t - \tilde{\mathbf{p}}_t)\right\| &\leq \frac{1}{1-\eta}\frac{1}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}\frac{1}{m}\sum_{i=1}^{m}\|(\mathbf{g}_{i,t} - \mathbf{g}_t)\|\\ &\leq \frac{1}{1-\eta}\frac{1}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}(1 + \sqrt{2\ln(\frac{m}{\delta})})\sqrt{\frac{1}{s}}\max_i\|\mathbf{b}_i\|. \end{aligned} \tag{16}$$

Now collecting the terms of (16) and (13) and plugging them into (12) we have

$$\begin{aligned}&\phi_t(\hat{\mathbf{p}}_t) - \phi_t(\mathbf{p}^*) \leq \epsilon^2 - \zeta^2\phi_t(\mathbf{p}^*)\\ &\Rightarrow \phi_t(\hat{\mathbf{p}}_t) \leq \epsilon^2 + (1-\zeta^2)\phi_t(\mathbf{p}^*),\end{aligned}$$

where ϵ is as defined in (4).

□

Lemma 6. *Let $\zeta \in (0,1), \epsilon$ be any fixed parameter. And $\hat{p}_t$ satisfies $\phi_t(\hat{\mathbf{p}}_t) \leq \epsilon^2 + (1-\zeta^2)\min_{\mathbf{p}} \phi_t(\mathbf{p})$. Under the Assumption 1(Hessian L-Lipschitz) and $\boldsymbol{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$ satisfies*

$$\boldsymbol{\Delta}_{t+1}^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} \leq L\|\boldsymbol{\Delta}_{t+1}\|\|\boldsymbol{\Delta}_t\|^2 + \frac{\zeta^2}{1-\zeta^2}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t + 2\epsilon^2.$$

Proof. We have $\mathbf{w}_{t+1} = \mathbf{w}_t - \hat{\mathbf{p}}_t$, $\boldsymbol{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$ and $\boldsymbol{\Delta}_{t+1} = \mathbf{w}_{t+1} - \mathbf{w}^*$. Also $\hat{\mathbf{p}}_t = \mathbf{w}_t - \mathbf{w}_{t+1} = \boldsymbol{\Delta}_t - \boldsymbol{\Delta}_{t+1}$. From the definition of ϕ we have,

$$\phi_t(\hat{\mathbf{p}}_t) = \frac{1}{2}(\boldsymbol{\Delta}_t - \boldsymbol{\Delta}_{t+1})^\top \mathbf{H}_t(\boldsymbol{\Delta}_t - \boldsymbol{\Delta}_{t+1}) - (\boldsymbol{\Delta}_t - \boldsymbol{\Delta}_{t+1}))\,\mathbf{g}_t,$$
$$(1-\zeta^2)\phi_t(\frac{1}{(1-\zeta^2)}\boldsymbol{\Delta}_t) = \frac{1}{2(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t - \boldsymbol{\Delta}_t^\top \mathbf{g}_t.$$

From the above two equation we have

$$\begin{aligned}
&\phi_t(\hat{\mathbf{p}}_t) - (1-\zeta^2)\phi_t(\frac{1}{(1-\zeta^2)}\boldsymbol{\Delta}_t) \\
&= \frac{1}{2}\boldsymbol{\Delta}_{t+1}^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} - \frac{1}{2}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} + \frac{1}{2}\boldsymbol{\Delta}_{t+1}^\top \mathbf{g}_t - \frac{\zeta^2}{2(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t.
\end{aligned}$$

From Lemma 5 the following holds

$$\begin{aligned}
\phi_t(\hat{\mathbf{p}}_t) &\leq \epsilon^2 + (1-\zeta^2)\min_{\mathbf{p}} \phi_t(\mathbf{p}) \\
&\leq \epsilon^2 + (1-\zeta^2)\phi_t(\frac{1}{(1-\zeta^2)}\boldsymbol{\Delta}_t).
\end{aligned}$$

So we have

$$\frac{1}{2}\boldsymbol{\Delta}_{t+1}^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} - \boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} + \boldsymbol{\Delta}_{t+1}^\top \mathbf{g}_t - \frac{\zeta^2}{2(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t \leq \epsilon^2. \tag{17}$$

Consider $\mathbf{g}_t = \mathbf{g}(\mathbf{w}_t)$

$$\begin{aligned}
\mathbf{g}(\mathbf{w}_t) &= \mathbf{g}(\mathbf{w}^*) + \left(\int_0^1 \nabla^2 f(\mathbf{w}^* + z(\mathbf{w}_t - \mathbf{w}^*))dz\right)(\mathbf{w}_t - \mathbf{w}^*) \\
&= \left(\int_0^1 \nabla^2 f(\mathbf{w}^* + z(\mathbf{w}_t - \mathbf{w}^*))dz\right)\boldsymbol{\Delta}_t \quad (\text{as } \mathbf{g}(\mathbf{w}^*) = 0).
\end{aligned}$$

Now we bound the following

$$\begin{aligned}
\|\mathbf{H}_t\boldsymbol{\Delta}_t - \mathbf{g}(\mathbf{w}_t)\| &\leq \|\boldsymbol{\Delta}_t\| \left\| \int_0^1 [\nabla^2 f(\mathbf{w}_t) - \nabla^2 f(\mathbf{w}^* + z(\mathbf{w}_t - \mathbf{w}^*))]dz \right\| \\
&\leq \|\boldsymbol{\Delta}_t\| \int_0^1 \left\| [\nabla^2 f(\mathbf{w}_t) - \nabla^2 f(\mathbf{w}^* + z(\mathbf{w}_t - \mathbf{w}^*))] \right\| dz \quad \text{(By Jensen's Inequality)} \\
&\leq \|\boldsymbol{\Delta}_t\| \int_0^1 (1-z)L\|\mathbf{w}_t - \mathbf{w}^*\| dz \quad \text{(by } L\text{-Lipschitz assumption)} \\
&= \frac{L}{2}\|\boldsymbol{\Delta}_t\|^2.
\end{aligned}$$

Plugging it into (17) we have

$$\begin{aligned}
\boldsymbol{\Delta}_{t+1}^\top \mathbf{H}_t \boldsymbol{\Delta}_{t+1} &\leq 2\boldsymbol{\Delta}_{t+1}^\top (\mathbf{H}_t\boldsymbol{\Delta}_t - \mathbf{g}_t) + \frac{\zeta^2}{(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t + 2\epsilon^2 \\
&\leq 2\|\boldsymbol{\Delta}_{t+1}\|\,\|\mathbf{H}_t\boldsymbol{\Delta}_t - \mathbf{g}_t\| + \frac{\zeta^2}{(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t + 2\epsilon^2 \\
&\leq L\|\boldsymbol{\Delta}_{t+1}\|\,\|\boldsymbol{\Delta}_t\|^2 + \frac{\zeta^2}{(1-\zeta^2)}\boldsymbol{\Delta}_t^\top \mathbf{H}_t \boldsymbol{\Delta}_t + 2\epsilon^2.
\end{aligned}$$

□

Proof of Theorem 1

Proof. From the Lemma 6 with probability $1-\delta$

$$\begin{aligned}\boldsymbol{\Delta}_{t+1}^{\top}\mathbf{H}_t\boldsymbol{\Delta}_{t+1} &\leq L\left\|\boldsymbol{\Delta}_{t+1}\right\|\left\|\boldsymbol{\Delta}_t\right\|^2+\frac{\zeta^2}{(1-\zeta^2)}\boldsymbol{\Delta}_t^{\top}\mathbf{H}_t\boldsymbol{\Delta}_t+2\epsilon^2\\ &\leq L\|\boldsymbol{\Delta}_{t+1}\|\|\boldsymbol{\Delta}_t\|^2+(\frac{\zeta^2}{1-\zeta^2}\sigma_{max}(\mathbf{H}_t))\|\boldsymbol{\Delta}_t\|^2+2\epsilon^2.\end{aligned}$$

So we have,

$$\|\boldsymbol{\Delta}_{t+1}\|\leq\max\{\sqrt{\frac{\sigma_{max}(\mathbf{H}_t)}{\sigma_{min}(\mathbf{H}_t)}(\frac{\zeta^2}{1-\zeta^2})}\|\boldsymbol{\Delta}_t\|,\frac{L}{\sigma_{min}(\mathbf{H}_t)}\|\boldsymbol{\Delta}_t\|^2\}+\frac{2\epsilon}{\sqrt{\sigma_{min}(\mathbf{H}_t)}}.$$

□

9 Appendix B: Analysis of Section 4

In this section we provide the theoretical analysis of the Byzantine robust method explained in Section 4 and prove the statistical guarantee. In any iteration t the following holds

$$\begin{aligned}|\mathcal{U}_t| &= |(\mathcal{U}_t\cap\mathcal{M}_t)|+|(\mathcal{U}_t\cap\mathcal{B}_t)|\\ |\mathcal{M}_t| &= |(\mathcal{U}_t\cap\mathcal{M}_t)|+|(\mathcal{M}_t\cap\mathcal{T}_t)|.\end{aligned}$$

Combining both we have

$$|\mathcal{U}_t| = |\mathcal{M}_t|-|(\mathcal{M}_t\cap\mathcal{T}_t)|+|(\mathcal{U}_t\cap\mathcal{B}_t)|.$$

Lemma 7. *Let $\{\mathbf{S}_i\}_{i=1}^m\in\mathbb{R}^{n\times s}$ be sketching matrices based on Lemma 2. Let ϕ_t be defined in* (10) *and $\hat{\mathbf{p}}_t$ be defined in Algorithm 1. It holds that*

$$\min_{\mathbf{p}}\phi_t(\mathbf{p})\leq\phi_t(\hat{\mathbf{p}}_t)\leq\epsilon_{byz}^2+(1-\zeta_{byz}^2)\phi(\mathbf{p}^*),$$

where ϵ_{byz} and ζ_{byz} is defined in (5) *and* (6) *respectively.*

Proof. In the following analysis we omit the subscript 't'. From the definition of the quadratic function (10) we know that

$$\phi(\hat{\mathbf{p}})-\phi(\mathbf{p}^*)=\frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}-\mathbf{p}^*)\|^2.$$

Now we consider

$$\begin{aligned}\frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}-\mathbf{p}^*)\|^2 &= \frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}}\hat{\mathbf{p}}_i-\mathbf{p}^*)\|^2\\ &= \frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\hat{\mathbf{p}}_i-\mathbf{p}^*)-\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\hat{\mathbf{p}}_i-\mathbf{p}^*)+\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\hat{\mathbf{p}}_i-\mathbf{p}^*))\|^2\\ &\leq \underbrace{\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\hat{\mathbf{p}}_i-\mathbf{p}^*)\|^2}_{Term1}+\underbrace{2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\hat{\mathbf{p}}_i-\mathbf{p}^*)\|^2}_{Term2}+\underbrace{2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\hat{\mathbf{p}}_i-\mathbf{p}^*))\|^2}_{Term3}.\end{aligned}$$

Now we bound each term separately and use the result of the Lemma 5 to bound each term.

$$\begin{aligned}Term1 &= \|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\hat{\mathbf{p}}_i-\mathbf{p}^*)\|^2\\ &= (\frac{1-\alpha}{1-\beta})^2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{M}|}(\sum_{i\in\mathcal{M}}(\hat{\mathbf{p}}_i-\mathbf{p}^*)\|^2\\ &\leq (\frac{1-\alpha}{1-\beta})^2[\epsilon^2+\zeta_{\mathcal{M}}^2\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2],\end{aligned}$$

where $\zeta_{\mathcal{M}} = \nu(\frac{\eta}{\sqrt{|\mathcal{M}|}} + \frac{\eta^2}{1-\eta}) = \nu(\frac{\eta}{\sqrt{(1-\alpha)m}} + \frac{\eta^2}{1-\eta})$.

Similarly the Term 2 can be bonded as it is a bound on good machines

$$\begin{aligned}
Term2 &= 2\|\mathbf{H}^{\frac{1}{2}} \frac{1}{|\mathcal{U}|} \sum_{i \in (\mathcal{M} \cap \mathcal{T})} (\hat{\mathbf{p}}_i - \mathbf{p}^*)\|^2 \\
&= 2(\frac{1-\alpha}{1-\beta})^2 \|\mathbf{H}^{\frac{1}{2}} \frac{1}{|\mathcal{M} \cap \mathcal{T}|} \sum_{i \in (\mathcal{M} \cap \mathcal{T})} (\hat{\mathbf{p}}_i - \mathbf{p}^*)\|^2 \\
&\leq 2(\frac{1-\alpha}{1-\beta})^2 [\epsilon^2 + \zeta^2_{\mathcal{M} \cap \mathcal{T}} \|\mathbf{H}^{\frac{1}{2}} \mathbf{p}^*\|^2],
\end{aligned}$$

where $\zeta_{\mathcal{M} \cap \mathcal{T}} = \nu(\frac{\eta}{\sqrt{|\mathcal{M} \cap \mathcal{T}|}} + \frac{\eta^2}{1-\eta}) \leq \nu(\frac{\eta}{\sqrt{(1-\beta)m}} + \frac{\eta^2}{1-\eta})$.

For the Term 3 we know that $\beta > \alpha$ so all the untrimmed worker norm is bounded by a good machine as at least one good machine gets trimmed.

$$\begin{aligned}
Term3 &= 2\|\mathbf{H}^{\frac{1}{2}} \frac{1}{|\mathcal{U}|} \sum_{i \in (\mathcal{U} \cap \mathcal{B})} (\hat{\mathbf{p}}_i - \mathbf{p}^*))\|^2 \\
&\leq 2\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \|\frac{1}{|\mathcal{U} \cap \mathcal{B}|} \sum_{i \in (\mathcal{U} \cap \mathcal{B})} (\hat{\mathbf{p}}_i - \mathbf{p}^*))\|^2 \\
&\leq 2\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \frac{1}{|\mathcal{U} \cap \mathcal{B}|} \sum_{i \in (\mathcal{U} \cap \mathcal{B})} \|(\hat{\mathbf{p}}_i - \mathbf{p}^*))\|^2 \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \frac{1}{|\mathcal{U} \cap \mathcal{B}|} \sum_{i \in (\mathcal{U} \cap \mathcal{B})} (\|\hat{\mathbf{p}}_i\|^2 + \|\mathbf{p}^*\|^2) \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \max_{i \in \mathcal{M}} (\|\hat{\mathbf{p}}_i\|^2 + \|\mathbf{p}^*\|^2) \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \max_{i \in \mathcal{M}} (\|\hat{\mathbf{p}}_i - \mathbf{p}^*\|^2 + 2\|\mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 \max_{i \in \mathcal{M}} (\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}_i - \mathbf{p}^*)\|^2 + 2\|\mathbf{H}^{\frac{1}{2}} \mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{|\mathcal{U} \cap \mathcal{B}|}{|\mathcal{U}|})^2 (\epsilon^2 + (2 + \zeta_1^2)\|\mathbf{H}^{\frac{1}{2}} \mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{\alpha}{1-\beta})^2 (\epsilon^2 + (2 + \zeta_1^2)\|\mathbf{H}^{\frac{1}{2}} \mathbf{p}^*\|^2),
\end{aligned}$$

where $\zeta_1 = \nu(\eta + \frac{\eta^2}{1-\eta}) = \frac{\nu}{1-\eta}$ and $\kappa = \frac{\sigma_{max}(\mathbf{H})}{\sigma_{min}(\mathbf{H})}$.

Combining all the bounds on Term1 , Term2 and Term3 we have

$$\frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}} - \mathbf{p}^*)\|^2 \leq \epsilon^2_{byz} + \zeta^2_{byz}\|\mathbf{H}^{\frac{1}{2}} \mathbf{p}^*\|^2,$$

where

$$\begin{aligned}
\epsilon^2_{byz} &= \left(3\left(\frac{1-\alpha}{1-\beta}\right)^2 + 4\kappa\left(\frac{\alpha}{1-\beta}\right)^2\right)\epsilon^2, \\
\zeta^2_{byz} &= 2\left(\frac{1-\alpha}{1-\beta}\right)^2 \zeta^2_{\mathcal{M} \cap \mathcal{T}} + \left(\frac{1-\alpha}{1-\beta}\right)^2 \zeta^2_{\mathcal{M}} + 4\kappa\left(\frac{\alpha}{1-\beta}\right)^2 (2 + \zeta_1^2).
\end{aligned}$$

Finally we have

$$\begin{aligned}
\phi(\hat{\mathbf{p}}) - \phi(\mathbf{p}^*) &\leq \epsilon^2_{byz} - \zeta^2_{byz}\phi(\mathbf{p}^*) \\
\Rightarrow \phi(\hat{\mathbf{p}}) &\leq \epsilon^2_{byz} + (1 - \zeta^2_{byz})\phi(\mathbf{p}^*).
\end{aligned}$$

□

Lemma 8. *Let $\zeta_{byz} \in (0,1), \epsilon_{byz}$ be any fixed parameter. And $\hat{p}_t$ satisfies $\phi_t(\hat{\mathbf{p}}_t) \leq \epsilon_{byz}^2 + (1 - \zeta_{byz}^2) \min_{\mathbf{p}} \phi_t(\mathbf{p})$. Under the Assumption 1(Hessian L-Lipschitz) and $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$ satisfies*

$$\mathbf{\Delta}_{t+1}^T \mathbf{H}_t \mathbf{\Delta}_{t+1} \leq L\|\mathbf{\Delta}_{t+1}\|\|\mathbf{\Delta}_t\|^2 + \frac{\zeta_{byz}^2}{1-\zeta_{byz}^2} \mathbf{\Delta}_t^T \mathbf{H}_t \mathbf{\Delta}_t + 2\epsilon_{byz}^2.$$

Proof. We choose $\zeta = \zeta_{byz}$ and $\epsilon = \epsilon_{byz}$ from the Lemma 7 and follow the proof of Lemma 6 to obtain the desired bound. □

Proof of Theorem 2

Proof. We get the desired bound by developing from the result of the Lemma 8 and following the proof of Theorem 1 □

10 Appendix C:Analysis of Section 5

First we prove the following lemma that will be useful in our subsequent calculations. Consider that $\mathcal{Q}(\hat{\mathbf{p}}) = \frac{1}{|B|} \sum_{i \in B} \mathcal{Q}(\hat{\mathbf{p}}_i)$. And also we use the following notation $\zeta_B = \nu(\frac{\eta}{\sqrt{|B|}} + \frac{\eta^2}{1-\eta})$, $\nu = \frac{\sigma_{max}(\mathbf{A}^\top \mathbf{A})}{\sigma_{max}(\mathbf{A}^\top \mathbf{A}) + n\lambda} \leq 1$.

Lemma 9. *If $\mathcal{Q}(\hat{\mathbf{p}}_i)$ is the local update direction and $\mathbf{p}^*$ is the optimal solution to the quadratic function ϕ then*

$$\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}_i) - \mathbf{p}^*)\right\|^2 \leq 1 + \kappa(1-\rho))\epsilon^2 + (\zeta_B^2 + \kappa(1-\rho)((1+\zeta_1^2)) \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2,$$

where $\mathbf{H}$ is the exact Hessian and

$$\epsilon_1 = \sqrt{(1+\kappa(1-\rho))}\epsilon,$$
$$\zeta_{comp,B}^2 = (\zeta_B^2 + \kappa(1-\rho)((1+\zeta_1^2)).$$

ϵ is defined in equation (4) *and*

Proof.

$$\begin{aligned}\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}) - \mathbf{p}^*)\right\|^2 &= \left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}) - \hat{\mathbf{p}} + \hat{\mathbf{p}} - \mathbf{p}^*)\right\|^2 \\ &\leq 2\left(\underbrace{\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}) - \hat{\mathbf{p}})\right\|^2}_{Term1} + \underbrace{\left\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}} - \mathbf{p}^*)\right\|^2}_{Term2}\right). \end{aligned} \tag{18}$$

Following the proof of Lemma 5 we get

$$\left\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}_i - \mathbf{p}^*)\right\|^2 \leq \epsilon^2 + \zeta_1 \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2, \tag{19}$$

where ϵ is as defined in (4).Now we consider the term

$$\begin{aligned}\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}_i) - \hat{\mathbf{p}}_i\right\|^2 &\leq \sigma_{max}(\mathbf{H})(1-\rho)\left\|\hat{\mathbf{p}}_i\right\|^2 \\ &\leq \sigma_{max}(\mathbf{H})(1-\rho)\left(\left\|\hat{\mathbf{p}}_i - \mathbf{p}^*\right\|^2 + \left\|\mathbf{p}^*\right\|^2\right) \\ &\leq \frac{\sigma_{max}}{\sigma_{min}}(1-\rho)\left(\left\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}_i - \mathbf{p}^*)\right\|^2 + \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2\right) \\ &= \kappa(1-\rho)\left(\left\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}_i - \mathbf{p}^*)\right\|^2 + \left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2\right) \\ &\leq \kappa(1-\rho)\left(\epsilon^2 + (1+\zeta_1^2)\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2\right) \quad \text{Using (19).}\end{aligned}$$

Now we use the above calculation and bound Term1

$$\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}})-\hat{\mathbf{p}})\right\|^2 \leq \frac{1}{|B|}\sum_{i\in B}\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\hat{\mathbf{p}}_i\right\|^2$$
$$\leq \kappa(1-\rho)\left(\epsilon^2+(1+\zeta_1^2)\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2\right). \tag{20}$$

We can bound the Term2 directly using the proof of Lemma 5

$$\left\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}-\mathbf{p}^*)\right\|^2 \leq \epsilon^2+\zeta_B^2\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2. \tag{21}$$

Now we use (20) and (21) and plug them in (18)

$$\left\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}})-\mathbf{p}^*)\right\|^2 \leq (1+\kappa(1-\rho))\epsilon^2+(\zeta_B^2+\kappa(1-\rho)((1+\zeta_1^2))\left\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\right\|^2.$$

Now we define

$$\epsilon_1 = \sqrt{(1+\kappa(1-\rho))}\epsilon$$
$$\zeta^2_{comp,B} = (\zeta_B^2+\kappa(1-\rho)((1+\zeta_1^2)).$$

□

Now we have the robust update in iteration t to be $\mathcal{Q}(\hat{\mathbf{p}}) = \frac{1}{|\mathcal{U}_t|}\sum_{i\in\mathcal{U}_t}\mathcal{Q}(\hat{p}_{i,t})$.

Lemma 10. *Let $\{\mathbf{S}_i\}_{i=1}^m \in \mathbb{R}^{n\times s}$ be sketching matrices based on Lemma 2. Let ϕ_t be defined in* (10) *and $\mathcal{Q}(\hat{\mathbf{p}}_t)$ be the update with $\mathcal{Q}$ being ρ-approximate compressor. It holds that*

$$\min_{\mathbf{p}}\phi_t(\mathbf{p}) \leq \phi_t(\mathcal{Q}(\hat{\mathbf{p}}_t)) \leq \epsilon^2_{comp,byz}+(1-\zeta^2_{comp,byz})\phi_t(\mathbf{p}^*),$$

where $\epsilon_{comp,byz}$ and $\zeta^2_{comp,byz}$ is as defined in (7) *and* (8) *respectively.*

Proof. In the following analysis we omit the subscript 't'. From the definition of the quadratic function (10) we know that

$$\phi(\mathcal{Q}(\hat{\mathbf{p}}))-\phi(\mathbf{p}^*) = \frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}})-\mathbf{p}^*)\|^2.$$

Now we consider

$$\begin{aligned}
\frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}})-\mathbf{p}^*)\|^2 &= \frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}}\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2\\
&= \frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)-\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)+\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*))\|^2\\
&\leq \underbrace{\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2}_{Term1}+\underbrace{2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2}_{Term2}\\
&+\underbrace{2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*))\|^2}_{Term3}.
\end{aligned}$$

Now we bound each term separately and use the Lemma 9

$$\begin{aligned}
Term1 &= \|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}(\sum_{i\in\mathcal{M}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2\\
&= (\frac{1-\alpha}{1-\beta})^2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{M}|}(\sum_{i\in\mathcal{M}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2\\
&\leq (\frac{1-\alpha}{1-\beta})^2[\epsilon_1^2+\zeta^2_{comp,\mathcal{M}}\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2],
\end{aligned}$$

where $\zeta^2_{comp,\mathcal{M}} = (\zeta^2_{\mathcal{M}} + \kappa(1-\rho)((1+\zeta_1^2)$. Similarly the Term 2 can be bonded as it is a bound on good machines

$$\begin{aligned}
Term2 &= 2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2 \\
&= 2(\frac{1-\alpha}{1-\beta})^2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{M}\cap\mathcal{T}|}\sum_{i\in(\mathcal{M}\cap\mathcal{T})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2 \\
&\leq 2(\frac{1-\alpha}{1-\beta})^2[\epsilon_1^2+\zeta^2_{comp,\mathcal{M}\cap\mathcal{T}}\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2].
\end{aligned}$$

For the Term 3 we know that $\beta > \alpha$ so all the untrimmed worker norm is bounded by a good machine as at least one good machine gets trimmed.

$$\begin{aligned}
Term3 &= 2\|\mathbf{H}^{\frac{1}{2}}\frac{1}{|\mathcal{U}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*))\|^2 \\
&\leq 2\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\|\frac{1}{|\mathcal{U}\cap\mathcal{B}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*))\|^2 \\
&\leq 2\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\frac{1}{|\mathcal{U}\cap\mathcal{B}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}\|(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*))\|^2 \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\frac{1}{|\mathcal{U}\cap\mathcal{B}|}\sum_{i\in(\mathcal{U}\cap\mathcal{B})}(\|\mathcal{Q}(\hat{\mathbf{p}}_i)\|^2+\|\mathbf{p}^*\|^2) \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\max_{i\in\mathcal{M}}(\|\mathcal{Q}(\hat{\mathbf{p}}_i)\|^2+\|\mathbf{p}^*\|^2) \\
&\leq 4\sigma_{max}(\mathbf{H})(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\max_{i\in\mathcal{M}}(\|\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*\|^2+2\|\mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2\max_{i\in\mathcal{M}}(\|\mathbf{H}^{\frac{1}{2}}(\mathcal{Q}(\hat{\mathbf{p}}_i)-\mathbf{p}^*)\|^2+2\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{|\mathcal{U}\cap\mathcal{B}|}{|\mathcal{U}|})^2(\epsilon_1^2+(2+\zeta_1^2)\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2) \\
&\leq 4\kappa(\frac{\alpha}{1-\beta})^2(\epsilon_1^2+(2+\zeta_1^2)\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2).
\end{aligned}$$

Combining all the bounds on Term1 , Term2 and Term3 we have

$$\frac{1}{2}\|\mathbf{H}^{\frac{1}{2}}(\hat{\mathbf{p}}-\mathbf{p}^*)\|^2 \leq \epsilon^2_{byz}+\zeta^2_{byz}\|\mathbf{H}^{\frac{1}{2}}\mathbf{p}^*\|^2,$$

where

$$\begin{aligned}
\epsilon^2_{comp,byz} &= \left(3\left(\frac{1-\alpha}{1-\beta}\right)^2+4\kappa\left(\frac{\alpha}{1-\beta}\right)^2\right)\epsilon_1^2 \\
\zeta^2_{comp,byz} &= 2\left(\frac{1-\alpha}{1-\beta}\right)^2\zeta^2_{comp,\mathcal{M}\cap\mathcal{T}}+\left(\frac{1-\alpha}{1-\beta}\right)^2\zeta^2_{comp,\mathcal{M}}+4\kappa\left(\frac{\alpha}{1-\beta}\right)^2(2+\zeta^2_{comp,1}).
\end{aligned}$$

Finally we have

$$\begin{aligned}
\phi(\hat{\mathbf{p}})-\phi(\mathbf{p}^*) &\leq \epsilon^2_{comp,byz}-\zeta^2_{comp,byz}\phi(\mathbf{p}^*) \\
\Rightarrow \phi(\hat{\mathbf{p}}) &\leq \epsilon^2_{comp,byz}+(1-\zeta^2_{comp,byz})\phi(\mathbf{p}^*).
\end{aligned}$$

□

Lemma 11. *Let $\zeta_{comp,byz} \in (0,1), \epsilon_{comp,byz}$ be any fixed parameter. And $\mathcal{Q}(\hat{p}_t)$ satisfies $\phi_t(\mathcal{Q}(\hat{p}_t)) \leq \epsilon^2_{byz}+(1-\zeta^2_{byz})\min_{\mathbf{p}}\phi_t(\mathbf{p})$. Under the Assumption 1(Hessian L-Lipschitz) and $\mathbf{\Delta}_t = \mathbf{w}_t - \mathbf{w}^*$ satisfies*

$$\mathbf{\Delta}_{t+1}^T\mathbf{H}_t\mathbf{\Delta}_{t+1} \leq L\|\mathbf{\Delta}_{t+1}\|\|\mathbf{\Delta}_t\|^2+\frac{\zeta^2_{comp,byz}}{1-\zeta^2_{comp,byz}}\mathbf{\Delta}_t^T\mathbf{H}_t\mathbf{\Delta}_t+2\epsilon^2_{comp,byz}.$$

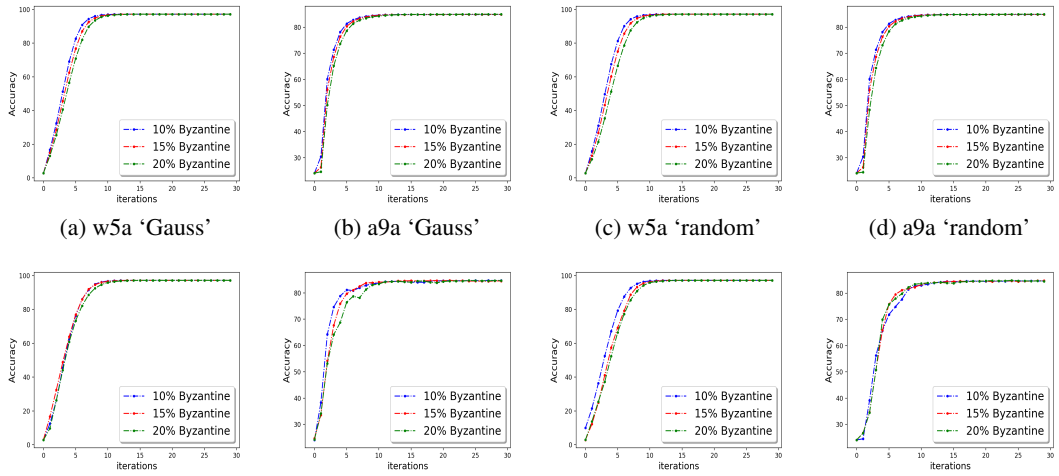
Proof. We choose $\zeta = \zeta_{comp,byz}$ and $\epsilon = \epsilon_{comp,byz}$ from the Lemma 10 and follow the proof of Lemma 6 to obtain the desired bound. □

Proof of Theorem 3

Proof. We get the desired bound by developing from the result of the Lemma 11 and following the proof of Theorem 1 □

11 Additional Experiment

In addition to the experimental results in Section 6, we provide some more experiments supporting the robustness of the COMRADE in two different types of attacks : 1. 'Gaussian attack': where the Byzantine workers add Gaussian Noise ($\mathcal{N}(\mu, \sigma^2)$) to the update and 2. 'random label attack': where the Byzantine worker machines learns based on random labels instead of proper labels.



(a) w5a 'Gauss' (b) a9a 'Gauss' (c) w5a 'random' (d) a9a 'random'

(e) w5a 'Gauss' (f) a9a 'Gauss' (g) w5a 'random' (h) a9a 'random'

Figure 3: (First row) Accuracy of COMRADE with $10\%, 15\%, 20\%$ Byzantine workers with 'Gaussian ' attack for (a). w5a (b). a9a and 'random label' attack for (c). w5a (d).a9a. (Second row) Accuracy of COMRADE with ρ-approximate compressor (Section 5) with $10\%, 15\%, 20\%$ Byzantine workers with 'Gaussian ' attack for (a). w5a (b). a9a and 'random label' attack for (c). w5a (d).a9a.