Distributed Newton-Type Methods with Communication Compression and Bernoulli Aggregation

Rustem Islamov[1] Xun Qian[2] Slavomír Hanzely[3] Mher Safaryan[3] Peter Richtárik[3]

[1]Insitut Polytechnique de Paris (IP Paris) [2]Shanghai Artificial Intelligence Laboratory [3]King Abdullah University of Science and Technology (KAUST)

The Problem and Assumptions

We want to solve the finite-sum optimization problem

$$\min_{x\in\mathbb{R}^d}\left\{ f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\} \qquad (1)$$

μ-strongly convex · # machines/devices · has Lipschitz Hessian · #model parameters · empirical loss/risk · local training data · local loss function $f_i(x) = \mathbb{E}_{\xi\sim\mathcal{D}_i}[f_\xi(x)]$

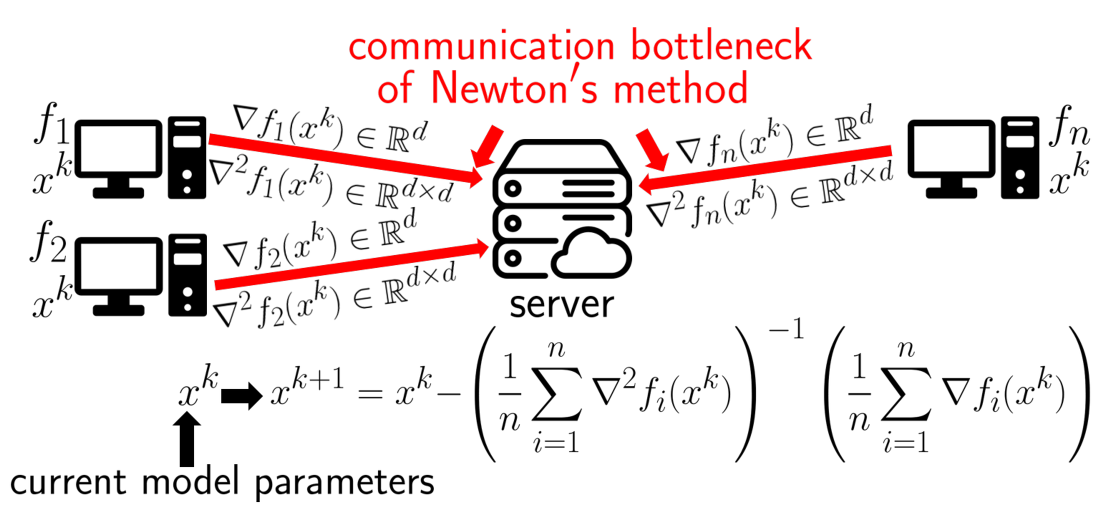
- Problem (1) has many applications in machine learning, data science and engineering.
- We focus on the regime when n **and** d **are very large**. This is typically the case in the big data settings (e.g., massively distributed and federated learning).

Notation: x^* is the unique solution of Problem (1).

Main goal

The goal is to create a Newton-type method supporting **communication compression** and **aggregation mechanisms** in order to reduce these costs while preserving theoretically superior local convergence guarantees. We aim to show that a wide variety of communication strategies, such as **contractive compression** and **lazy aggregation** can be applied on Hessian communication utilizing a class of **three point compressors (3PC)** [1].

Communication bottleneck



Learning Mechanism: the Idea

Newton-Star: $x^{k+1} = x^k - \nabla^2 f(x^*)\nabla f(x^k)$

- It converges locally quadratically;
- The communication cost of one iteration is $\mathcal{O}(d)$;
- It cannot be implemented in practice.

Following the idea of FedNL [2], in Newton-3PC we maintain a sequence of matrices $\mathbf{H}_i^k \in \mathbb{R}^{d\times d}$ with the goal of learning $\nabla^2 f_i(x^*)$. Then we can estimate the Hessian $\nabla^2 f(x^*)$ via $\nabla^2 f_i(x^*) \approx \mathbf{H}^k := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^k$.

Learning Mechanism: We update each local estimate $\mathbf{H}_i^{k+1}$ based on

- current local Hessian $\nabla^2 f_i(x^{k+1})$;
- previous local Hessian $\nabla^2 f_i(x^k)$;
- previous estimate $\mathbf{H}_i^{k+1}$

using the so-called 3PC compressor:

$$\mathbf{H}_i^{k+1} = \mathcal{C}_{\mathbf{H}_i^k,\nabla^2 f_i(x^k)}(\nabla^2 f_i(x^{k+1})).$$

Main features of the family of Newton- methods important for Federated Learning

- supports **heterogeneous data** setting
- uses **adaptive stepsizes**
- supports a wide range of **aggregation mechanisms** (e.g., LAG, BAG [2])
- fast local rate: **independent of the condition number**
- supports smart **uplink gradient compression** at the devices
- applies to general **finite-sum problems**
- **privacy is enhanced** (training data is not sent to the server)
- supports **3PC compressors** (e.g., EF21, CLAG, CBAG [2])
- supports **partial participation**
- supports smart **downlink model compression** by the server

Class of 3PC Compressors

Definition: We say that a (possibly randomized) map

$$\mathcal{C}_{\mathbf{H},\mathbf{Y}}(\mathbf{X}) : \underbrace{\mathbb{R}^{d\times d}}_{\mathbf{H}\in} \times \underbrace{\mathbb{R}^{d\times d}}_{\mathbf{Y}\in} \times \underbrace{\mathbb{R}^{d\times d}}_{\mathbf{X}\in} \mapsto \mathbb{R}^{d\times d} \qquad (2)$$

is a three point compressor (3PC) if there exist constants $0 < A \le 1$ and $B \ge 0$ such that

$$\mathbb{E}\left[\|\mathcal{C}_{\mathbf{H},\mathbf{Y}}(\mathbf{X}) - \mathbf{X}\|_{\rm F}^2\right] \le (1-A)\|\mathbf{H}-\mathbf{Y}\|_{\rm F}^2 + B\|\mathbf{X}-\mathbf{Y}\|_{\rm F}^2. \qquad (3)$$

holds for all matrices $\mathbf{H},\mathbf{Y},\mathbf{X} \in \mathbb{R}^{d\times d}$.

We use 3PC compressor with $\mathbf{Y} = \nabla^2 f_i(x^k)$, $\mathbf{H} = \mathbf{H}_i^k$, and $\mathbf{X} = \nabla^2 f_i(x^{k+1})$.

Example (Contractive compressors): The (possibly randomized) map $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ is called contractive compressor with contraction parameter $\alpha \in (0,1]$, if the following holds for any matrix $\mathbf{X} \in \mathbb{R}^{d\times d}$

$$\mathbb{E}\left[\|\mathcal{C}(\mathbf{X}) - \mathbf{X}\|_{\rm F}^2\right] \le (1-\alpha)\|\mathbf{X}\|_{\rm F}^2. \qquad (4)$$

Example (Compressed Lazy AGgregation (CLAG): Let $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ be a contractive compressor with contraction parameter $\alpha \in (0,1]$ and $\zeta \ge 0$ be a trigger for the aggregation. Then CLAG mechanism is defined as

$$\mathcal{C}_{\mathbf{H},\mathbf{Y}}(\mathbf{X}) = \begin{cases} \mathbf{H} + \mathcal{C}(\mathbf{X}-\mathbf{H}) & \text{if } \|\mathbf{X}-\mathbf{H}\|_{\rm F}^2 > \zeta\|\mathbf{X}-\mathbf{Y}\|_{\rm F}^2, \\ \mathbf{H} & \text{otherwise.} \end{cases} \qquad (5)$$

Example (Compressed Bernoulli AGgregation (CBAG) Let $\mathcal{C}\colon \mathbb{R}^d \to \mathbb{R}^d$ be a contractive compressor with contraction parameter $\alpha \in (0,1]$ and $p \in (0,1]$ be the probability for the aggregation. We then define CBAG mechanism is defined as

$$\mathcal{C}_{\mathbf{H},\mathbf{Y}}(\mathbf{X}) = \begin{cases} \mathbf{H} + \mathcal{C}(\mathbf{X}-\mathbf{H}) & \text{with probability } p, \\ \mathbf{H} & \text{with probability } 1-p. \end{cases} \qquad (6)$$

Communication advantages: Due to the *adaptive nature* of CLAG (5), Newton-CLAG method *does not send any information* about the local Hessian $\nabla^2 f_i(x^{k+1})$ if it is sufficiently close to previous Hessian estimate $\mathbf{H}_i^k$. It *reuses* local Hessian estimate $\mathbf{H}_i^k$ while there is no essential discrepancy.

Computation advantages: in CBAG (6) *probabilistic switching condition* is used according to Bernoulli random variable. This allows devices to compute local Hessian $\nabla^2 f_i(x^{k+1})$ and communicate compressed difference $\mathcal{C}(\nabla^2 f_i(x^{k+1}) - \mathbf{H}_i^k)$ only *with probability* p.

Algorithm 1: Newton-3PC

Input: $x^0 \in \mathbb{R}^d$, $\mathbf{H}_1^0,\dots,\mathbf{H}_n^0 \in \mathbb{R}^{d\times d}$, $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$, $l^0 = \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^0 - \nabla^2 f_i(x^0)\|_{\rm F}$

on server
- *Option 1:* $x^{k+1} = x^k - [\mathbf{H}^k]_\mu^{-1}\nabla f(x^k)$
- *Option 2:* $x^{k+1} = x^k - [\mathbf{H}^k + l^k\mathbf{I}]^{-1}\nabla f(x^k)$
- Broadcast x^{k+1} to all nodes

for *each device* $i = 1,\dots,n$ *in parallel* **do**
- Get x^k from the server and compute local gradient $\nabla f_i(x^k)$ and local Hessian $\nabla^2 f_i(x^k)$
- Apply 3PC and update local Hessian estimator to $\mathbf{H}_i^{k+1} := \mathcal{C}_{\mathbf{H}_i^k,\nabla^2 f_i(x^k)}(\nabla^2 f_i(x^{k+1}))$
- Send $\nabla f_i(x^{k+1})$, $\mathbf{H}_i^{k+1}$ and $l_i^{k+1} := \left\|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^{k+1})\right\|_{\rm F}$

end

on server
- Aggregate $\nabla f(x^{k+1}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^{k+1})$
- $\mathbf{H}^{k+1} = \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^{k+1}$, $l^{k+1} = \frac{1}{n}\sum_{i=1}^n l_i^{k+1}$

$[\cdot]_\mu$ denotes the projection onto the cone of positive definite matrices with eigenvalues bigger than μ.

Convergence Theory

Assumption: The average loss f is μ-strongly convex, and all local losses $f_i(x)$ have Lipschitz continuous Hessians:

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \le L_*\|x-y\|,$$
$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\|_{\rm F} \le L_{\rm F}\|x-y\|,$$
$$\max_{j,l}|(\nabla^2 f_i(x) - \nabla^2 f_i(y))_{jl}| \le L_\infty\|x-y\|$$

to hold for all $i \in [n]$ and $x,y \in \mathbb{R}^d$.

Notation: $(C,D) = \begin{cases} (2, L_*^2) & \text{if Option 1 is used} \\ (8, (L_*+2L_{\rm F})^2) & \text{if Option 2 is used} \end{cases}$

Define the Lyapunov function:

$$\Phi^k := \underbrace{\frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2}_{:=\mathcal{H}^k} + 6\left(\frac{1}{A} + 3AB\right)L_{\rm F}^2\|x^k - x^*\|^2.$$

Theorem: Assume $\|x^0 - x^*\| \le \frac{\mu}{\sqrt{2D}}$ and $\mathcal{H}^k \le \frac{\mu^2}{4C}$ for all $k \ge 0$. Then, Newton-3PC (Algorithm 1) with any 3PC mechanism converges with the following rates:

$$\|x^k - x^*\|^2 \le \frac{1}{2^k}\|x^0 - x^*\|^2, \qquad \mathbb{E}\left[\Phi^k\right] \le \left(1 - \min\left\{\frac{A}{2},\frac{1}{3}\right\}\right)^k \Phi^0,$$

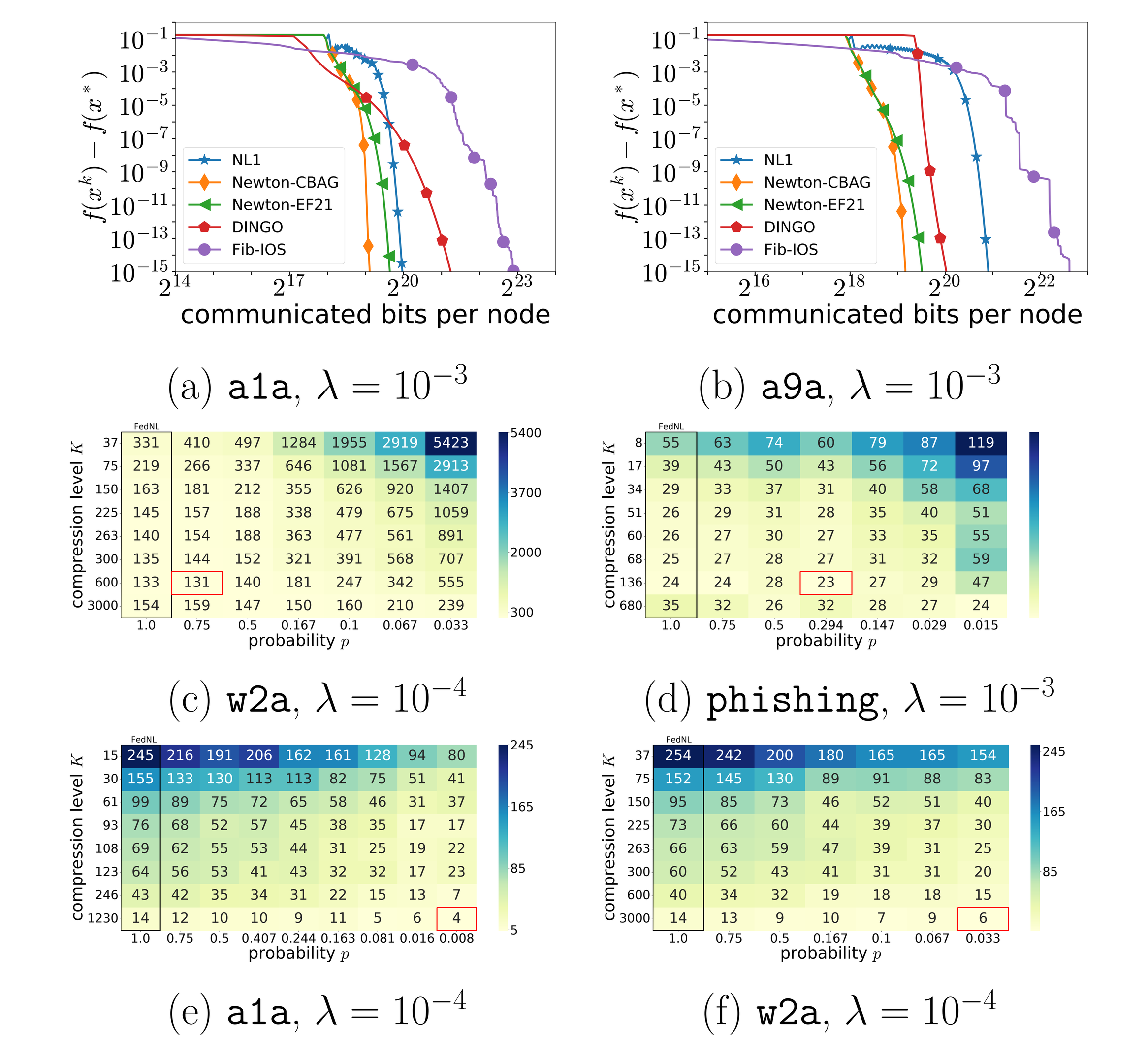
$$\mathbb{E}\left[\frac{\|x^{k+1}-x^*\|^2}{\|x^k - x^*\|^2}\right] \le \left(1 - \min\left\{\frac{A}{2},\frac{1}{3}\right\}\right)^k \left(C + \frac{AD}{12(1+3AB)L_{\rm F}^2}\right)\frac{\Phi^0}{\mu^2}.$$

Remark: In fact, assumption $\mathcal{H}^k \le \frac{\mu^2}{4C}$ for all $k \ge 0$ should hold only for $k = 0$.

Experiments

We consider L2 regularized logistic regression problem:

$$\min_{x\in\mathbb{R}^d}\left\{\frac{1}{n}\sum_{i=1}^n f_i(x) + \frac{\lambda}{2}\|x\|^2\right\}, \quad f_i(x) = \frac{1}{m}\sum_{j=1}^m \log\left(1 + e^{-b_{ij}a_{ij}^\top x}\right).$$



(a) a1a, $\lambda = 10^{-3}$ (b) a9a, $\lambda = 10^{-3}$ (c) w2a, $\lambda = 10^{-4}$ (d) phishing, $\lambda = 10^{-3}$ (e) a1a, $\lambda = 10^{-4}$ (f) w2a, $\lambda = 10^{-4}$

Figure 1: **First row:** Local comparison of NL1, Newton-CBAG, Newton-EF21 (equivalent to FedNL), DINGO, and Fib-IOS; **Second row:** The performance of Newton-CBAG combined with Top-K in terms of communication complexity (in Mbytes); **Third row:** The number of local Hessian computations of Newton-CBAG combined with Top-K.

References

[1] Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback, *Conference on Neural Information Processing Systems*, 2021.

[2] Mher Safaryan, Rustem Islamov, Xun Qian, Peter Richtárik. FedNL: Making Newton-Type Methods Applicable to Federated Learning, *International Conference on Machine Learning*, 2022.

[3] Xun Qian, Rustem Islamov, Mher Safaryan, and Peter Richtárik. Basis matters: Better communication-efficient second order methods for federated learning, *International Conference on Artificial Intelligence and Statistics*, 2022.

[4] Rustem Islamov, Xun Qian, and Peter Richtárik. Distributed Second Order Methods with Fast Rates and Compressed Communication, *International Conference on Machine Learning*, 2021.