Distributed Newton-Type Methods with Communication Compression and Bernoulli Aggregation

Department of Intelligent Systems
Moscow Institute of Physics and Technology
Master’s Thesis Defense

Rustem Islamov
Scientific advisor: Dr. Strijov V.V.

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Outline

The Problem

Strongly convex

clients/devices

2nd order smooth

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

model parameters

Empirical risk/loss

Local training data

Local loss function $f_i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i} [f_\xi(x)]$

Outline

1. **The Problem**
2. **Brief Comparison with Related Works**
3. The 3 Special Newton-type Methods
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Existing Approaches and their Disadvantages

First order methods

- ✖ Rates depend on the condition number
- ✖ Hard to find optimal stepsizes

Second order methods

- ✖ Rates depend on the condition number
- ✖ Communication cost is high

Recently, [Islamov et al, 2021], [Qian et al., 2022], [Safaryan et al., 2022] have developed new theory of second order methods for Federated and Distributed Learning solving most of the existing issues.

GOAL

1. Unify theory of these works into one creating more general class of compression operators

2. Improve the computational cost of the methods

Outline

1. The Problem
2. Brief Comparison with Related Works
3. Ideal Newton-type Method
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Newton Star Method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*)\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k)\right)$$

Can NOT be computed locally

Single communication of $\mathcal{O}(d^2)$

Can be computed locally

Easy to communicate: $\mathcal{O}(d)$

Rustem Islamov, Xun Qian and Peter Richtárik
Distributed second order methods with fast rates and compressed communication,
ICML 2021.

- $\mathcal{O}(d)$ communication cost per round
- Implementability in practice
- Local quadratic convergence rate independent of the condition number

Hessian Lipschitz constant

$$\|x^{k+1} - x^*\| \leq \frac{L}{2\mu}\|x^k - x^*\|^2$$

Strong convexity constant

Local quadratic rate

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Learning the Optimal Hessian Matrices

Newton Star

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*)\right)^{-1} \nabla f(x^k)$$

Idea! Learn the optimal Hessians $\nabla^2 f_i(x^*)$ in communication efficient manner:

(i) $\mathbf{H}_i^k \to \nabla^2 f_i(x^*)$ as $k \to \infty$ $\quad$ (ii) $\mathbf{H}_i^{k+1} - \mathbf{H}_i^k$ is compressed

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{H}_i^k\right)^{\boxed{-1}} \nabla f(x^k)$$

$$= x^k - \left(\mathbf{H}^k\right)^{\boxed{-1}} \nabla f(x^k)$$

Rustem Islamov, Xun Qian and Peter Richtárik
Distributed second order methods with fast rates and compressed communication,
ICML 2021.

Newton-3PC: Two Options for Updating the Global Model

Option 1

$$x^{k+1} = x^k - \left(\left[\mathbf{H}^k \right]_{\boxed{\mu}} \right)^{-1} \nabla f(x^k)$$

Projection onto the cone of positive definite matrices

Option 2

$$x^{k+1} = x^k - \left(\mathbf{H}^k + \boxed{l^k} \mathbf{I} \right)^{-1} \nabla f(x^k)$$

$$l^k = \frac{1}{n} \sum_{i=1}^{n} \| \mathbf{H}_i^k - \nabla^2 f_i(x^k) \|_{\mathrm{F}}$$

Newton-3PC: New Hessian Learning Technique

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

3PC compressor

Definition: 3PC compressor

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

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

3PC compressor: Examples

Contractive compressor

$\mathbf{H} \equiv \mathbf{0}$

EF21 compressor

$p = 1$

CBAG compressor

$\zeta = 0$

CLAG compressor

$$\mathbb{E}\left[\|\mathcal{C}(\mathbf{X}) - \mathbf{X}\|^2\right] \leq (1-\alpha)\|\mathbf{X}\|^2$$

$$\alpha \in (0, 1]$$

$$\begin{pmatrix} 1 & -2 \\ 1.9 & -0.5 \end{pmatrix} \xrightarrow{\text{Top-1}} \begin{pmatrix} 0 & -2 \\ 0 & 0 \end{pmatrix}$$

Top-K satisfies this with $\alpha = \frac{K}{d^2}$

$$\mathcal{C}_{\mathbf{H}}(\mathbf{X}) = \mathbf{H} + \mathcal{C}(\mathbf{X} - \mathbf{H})$$

Contractive compressor

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

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

Newton-3PC: Pseudocode

Algorithm 1 Newton-3PC (Newton's method with three point compressor)

Input: $x^0 \in \mathbb{R}^d$, $\mathbf{H}_1^0, \dots, \mathbf{H}_n^0 \in \mathbb{R}^{d\times d}$, $\mathbf{H}^0 := \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^0$, $l^0 = \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^0 - \nabla^2 f_i(x^0)\|_{\mathrm{F}}$.
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+1} and compute local gradient $\nabla f_i(x^{k+1})$ and local Hessian $\nabla^2 f_i(x^{k+1})$
    Apply 3PC and update local Hessian estimator to $\mathbf{H}_i^{k+1} = \mathcal{C}_{\mathbf{H}_i^k, \nabla^2 f_i(x^k)}\left(\nabla^2 f_i(x^{k+1})\right)$
    Send $\nabla f_i(x^{k+1})$, $\mathbf{H}_i^{k+1}$ and $l_i^{k+1} := \|\mathbf{H}_i^{k+1} - \nabla^2 f_i(x^{k+1})\|_{\mathrm{F}}$ to the server
end for
on server
    Aggregate $\nabla f(x^{k+1}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x^{k+1}), \mathbf{H}^{k+1} = \frac{1}{n}\sum_{i=1}^n \mathbf{H}_i^{k+1}, l^{k+1} = \frac{1}{n}\sum_{i=1}^n l_i^{k+1}$

Newton-3PC: Assumptions

1. For a given input $x \in \mathbb{R}^d$, clients can compute gradient $\nabla f_i(x)$ and Hessian $\nabla^2 f_i(x)$.
2. The average loss function $f(x)$ is μ-strongly convex for some $\mu \geq 0$.
3. Local loss functions $f_i(x)$ have Lipschitz continuous Hessians with constant $L \geq 0$, i.e.,

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq L\|x - y\|$$

   holds for any $x, y \in \mathbb{R}^d$.

Newton-3PC: Local Convergence Theory

Local rate

$$\|x^0 - x^*\| \leq \frac{\mu}{\sqrt{D}}$$

$$\|x^k - x^*\|^2 \boxed{\leq} \frac{1}{2^k}\|x^0 - x^*\|^2$$

Local (fixed) linear rate

Lyapunov function

$$\Phi^k := \mathcal{H}^k + 6\left(\frac{1}{A} + 3AB\right) L_{\mathrm{F}}^2 \left\|x^k - x^*\right\|^2$$

where $\mathcal{H}^k := \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\mathrm{F}}^2$

$$\mathbb{E}\left[\boxed{\Phi^k}\right] \boxed{\leq} \left(1 - \min\left\{\frac{\boxed{A}}{2}, \frac{1}{3}\right\}\right)^k \boxed{\Phi^0}$$

Local linear rates

$x^k \to x^*$

$\mathbf{H}_i^k \to \nabla^2 f_i(x^*)$ for all $i \in [n]$

Constant depending on the choice of the compressor and stepsize

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

Local superlinear rate

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Special Compressors to Save Time

CBAG compressor

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

We need to compute X only with probability p

Sketch&Project

$$\mathcal{C}(\mathbf{X}) = \mathbf{S}(\mathbf{S}^\top \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{X} \quad \text{where } \mathbf{S} \in \mathbb{R}^{d \times \tau} \sim \mathcal{D} \text{ and } \tau \ll d$$

We need to compute Hessian-vector products only

Outline

1. The Problem
2. Brief Comparison with Related Works
3. The 3 Special Newton-type Methods
4. Newton-3PC
5. Practical Extensions
6. Numerical Experiments

Experiments: Regularized Logistic Regression

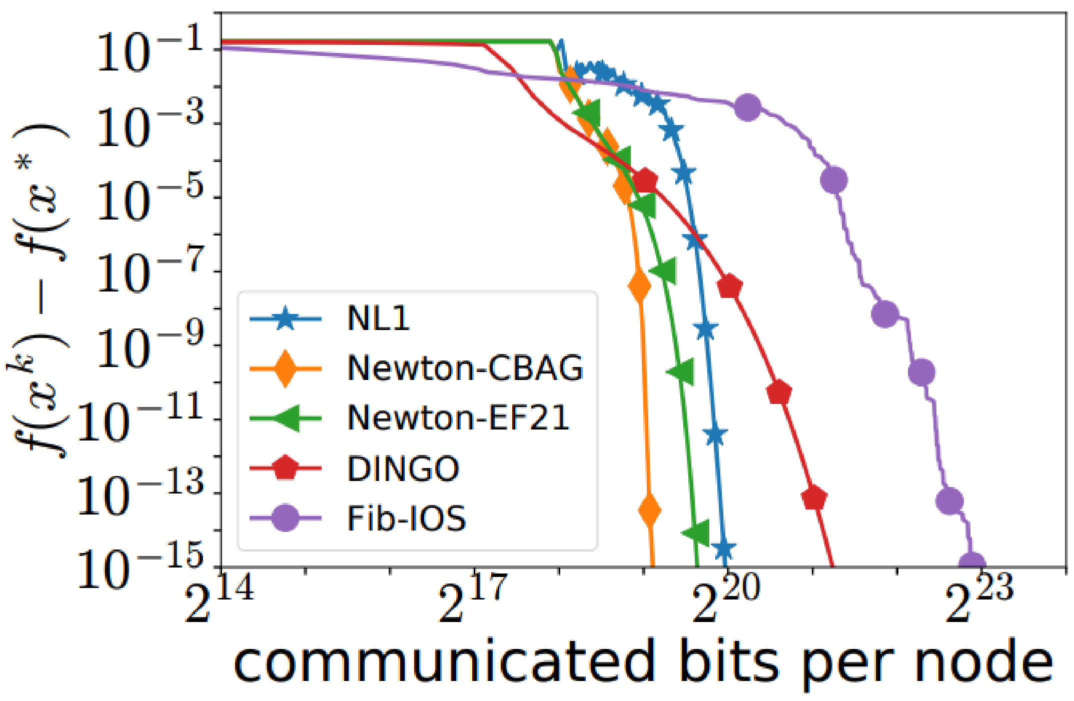
Regularization parameter

Training data points

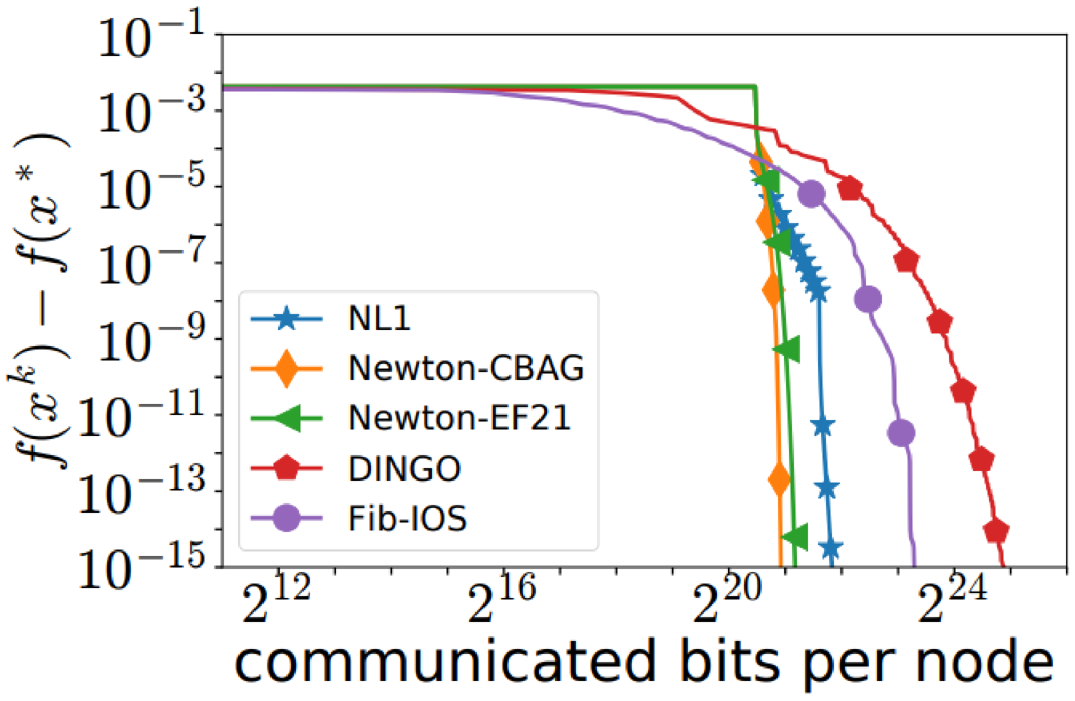
$$\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \frac{\lambda}{2}\|x\|^2\right\}, \qquad f_i(x) = \frac{1}{m}\sum_{j=1}^{m}\log\left(1+\exp(-b_{ij}a_{ij}^\top x)\right),$$

where $\{a_{ij}, b_{ij}\}_{j\in[m]}$ are data points at the i-th device. The datasets were taken from LibSVM library

Experiments: Local Comparison against Second Order Methods



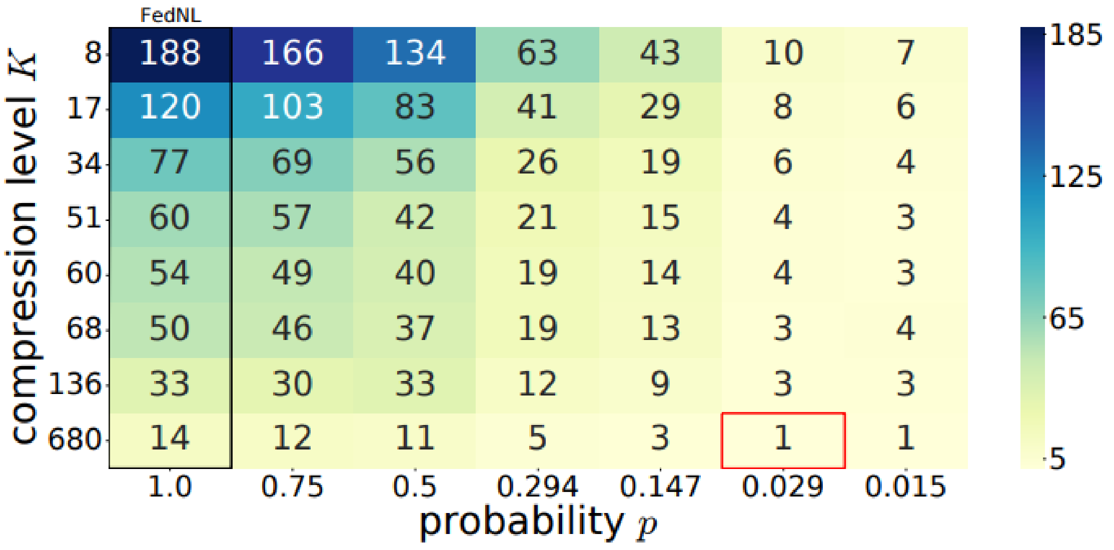
a1a, $\lambda = 10^{-3}$



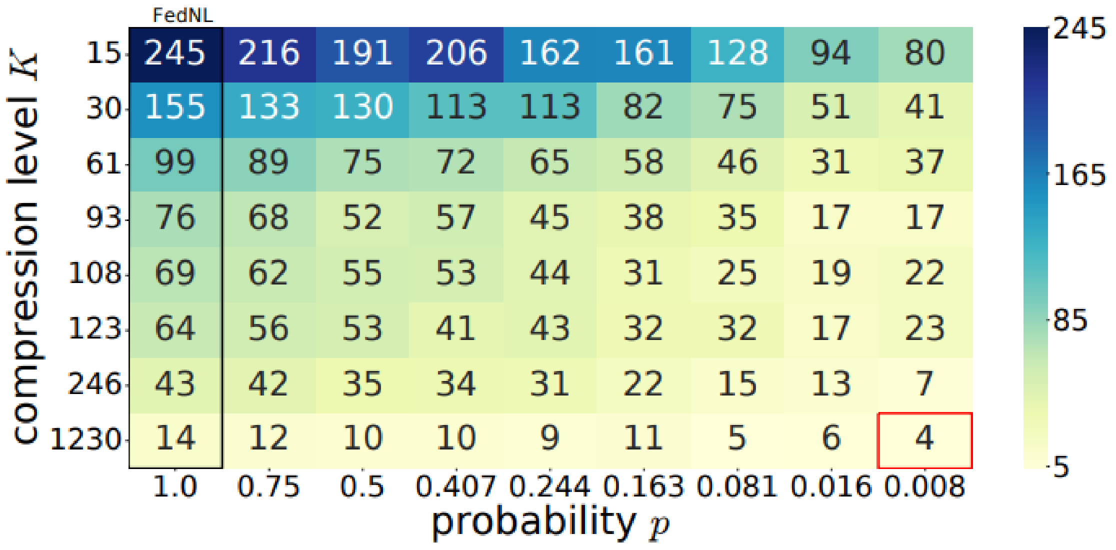
w2a, $\lambda = 10^{-4}$

We observe that Newton-CBAG outperforms other second-order methods in terms of communication complexity (some of them by several order in magnitude!). We also observe that randomization process of CBAG (p=1/d) improves on its deterministic version EF21 (p=1).

Experiments: CBAG indeed beneficial



$\texttt{phishing}, \lambda = 10^{-3}$



$\texttt{a1a}, \lambda = 10^{-4}$

We use CBAG compressor based on Top-K contractive compressor and vary compression level K and probability p.

We observe that smaller p leads to better communication complexity, which means that we do not have to compute Hessians in each iteration.

Thank you