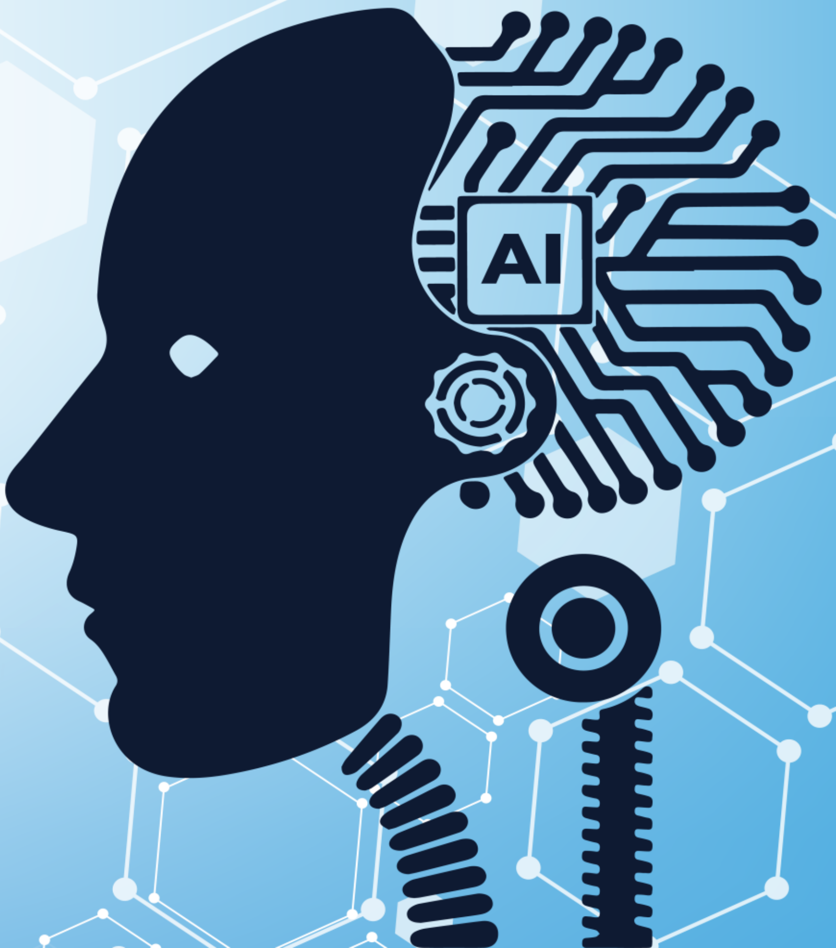
Distributed Second Order Methods with Fast Rates and Compressed Communication

Maths & AI: MIPT-UGA
young researchers workshop

Rustem Islamov

MIPT Inria
UGA Université Grenoble Alpes
MIAI Grenoble Alpes
LJK



Authors

Xun Qian
KAUST

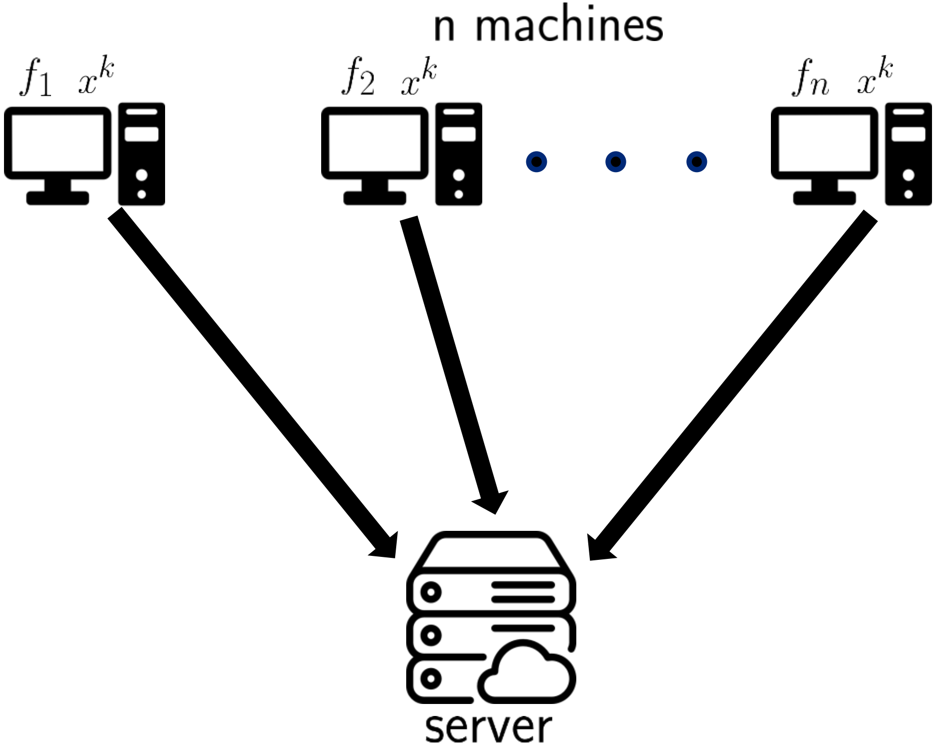
Peter Richtárik
KAUST

Rustem Islamov, Xun Qian and Peter Richtárik
Distributed Second Order Methods with Fast Rates and Compressed Communication
Accepted to International Conference on Machine Learning 2021
arXiv:2102.07158, 2021

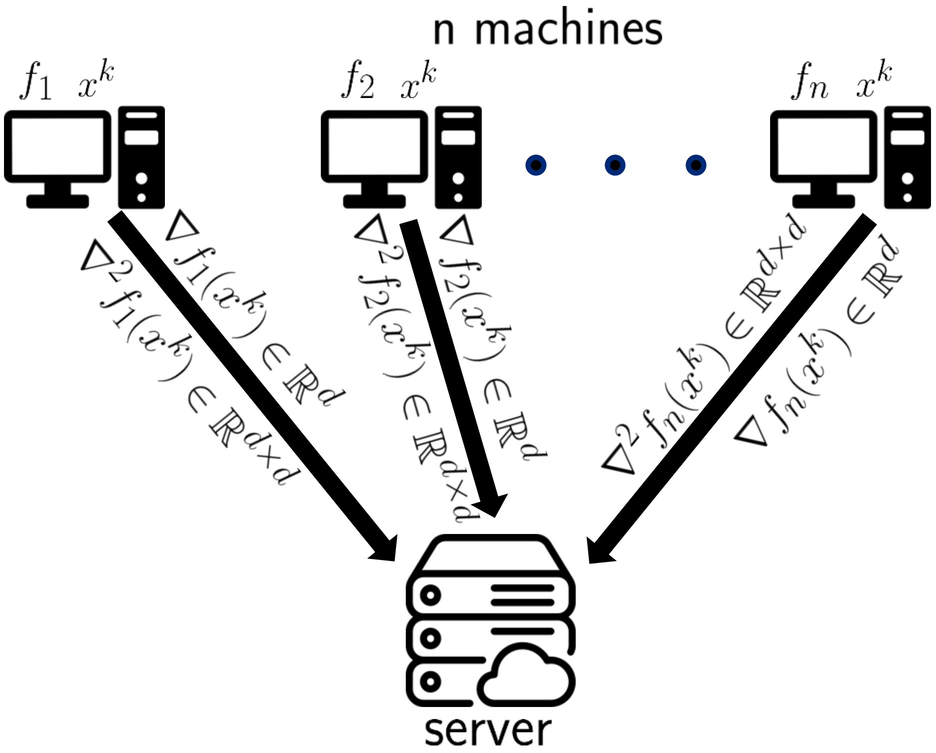
Outline

1. Distributed Optimization
2. Motivation
3. Newton's method
4. Newton Star
5. Newton Learn
6. Further results
7. Experiments

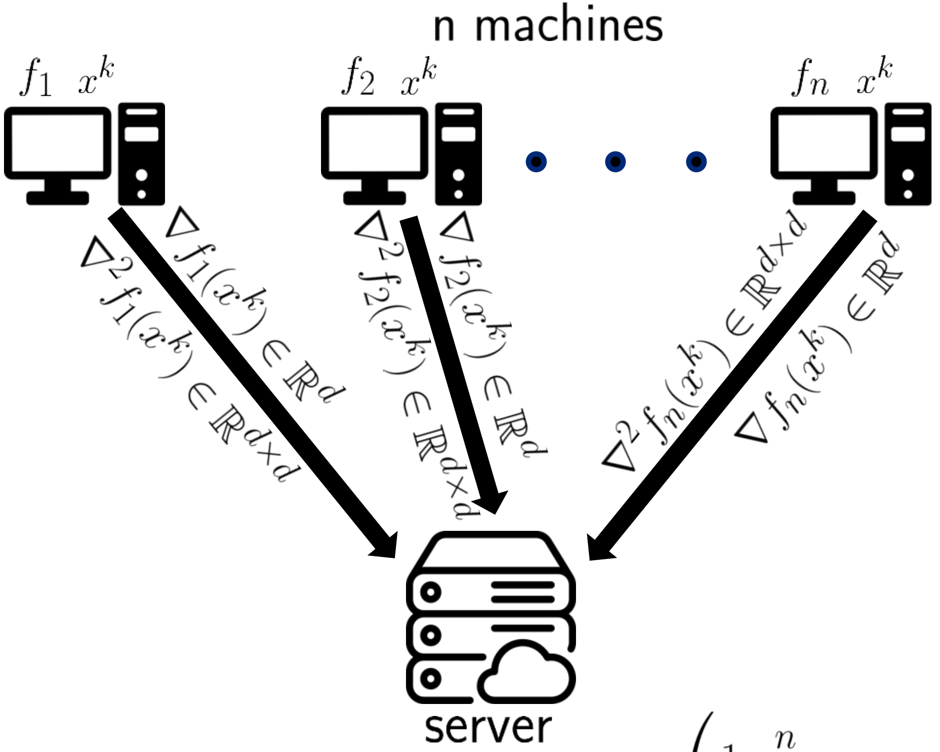
Server-Client Architecture



Server-Client Architecture

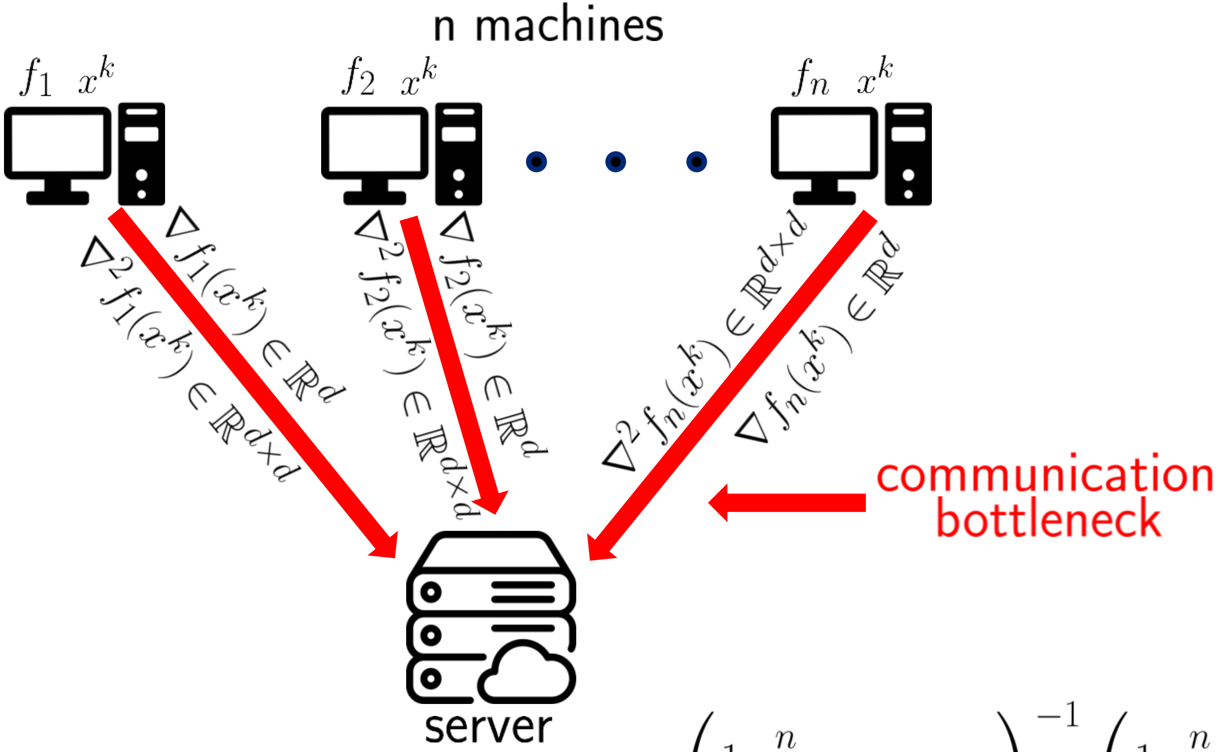


Server-Client Architecture



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

Server-Client Architecture



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

How to handle communication bottleneck?

Local methods

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma \nabla f_i(x^k) & \text{if } k \bmod T \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma \nabla f_i(x^k)\right) & \text{otherwise} \end{cases}$$

How to handle communication bottleneck?

Local methods

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma \nabla f_i(x^k) & \text{if } k \bmod T \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma \nabla f_i(x^k)\right) & \text{otherwise} \end{cases}$$

Examples: FedAvg, SCAFFOLD, Local-GD

Decrease the number of communication rounds

How to handle communication bottleneck?

Local methods

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma \nabla f_i(x^k) & \text{if } k \bmod T \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma \nabla f_i(x^k)\right) & \text{otherwise} \end{cases}$$

Examples: FedAvg, SCAFFOLD, Local-GD

Decrease the number of communication rounds

Compression

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

How to handle communication bottleneck?

Local methods

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma \nabla f_i(x^k) & \text{if } k \bmod T \neq 0 \\ \frac{1}{n}\sum_{i=1}^{n}\left(x_i^k - \gamma \nabla f_i(x^k)\right) & \text{otherwise} \end{cases}$$

Examples: FedAvg, SCAFFOLD, Local-GD

Decrease the number of communication rounds

Compression

$$x^{k+1} = x^k - \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathcal{C}_i^k\left(\nabla f_i(x^k)\right)}_{\text{Requires less bits}}$$

Examples: QSGD, DCGD, DIANA, ADIANA, MARINA

Pros and Cons of Distributed First order methods

Compressed methods

- ✅ Very well investigated already
- ✅ Provably benefit from compressed communication
- ❌ Rates depend on the condition number
- ❌ Hard to find optimal stepsizes

Examples: QSGD, DCGD, DIANA, ADIANA, MARINA

Pros and Cons of Distributed First order methods

Compressed methods

- ✅ Very well investigated already
- ✅ Provably benefit from compressed communication
- ❌ Rates depend on the condition number
- ❌ Hard to find optimal stepsizes

Examples: QSGD, DCGD, DIANA, ADIANA, MARINA

Local methods

- ✅ Not that well understood
- ✅ Very limited communication avoidance effect
- ❌ Rates depend on the condition number
- ❌ Hard to find optimal stepsizes

Bad for heterogeneous data

Second Order Methods to the Rescue?

Existing second order methods **suffer from at least one of these issues:**

- ❌ Communication cost is high (communication of Hessian matrices)
- ❌ Rates depend on the condition number
- ✅ Often no problem with stepsize selection

Second Order Methods to the Rescue?

Existing second order methods **suffer from at least one of these issues:**

- ✗ Communication cost is high (communication of Hessian matrices)
- ✗ Rates depend on the condition number
- ✓ Often no problem with stepsize selection

GOAL

Develop a communication-efficient distributed Newton-type method whose (local) convergence rate is independent of the condition number

- ✓ Can provably benefit from communication compression
- ✓ Rate is independent of the condition number
- ✗ Good rate for local convergence only
- ✓ No issue with stepsize selection
- ✓ New nature of local steps

The Problem

machines

training data points on each machine

$$\min_{x \in \mathbb{R}^d} \left\{ \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x) \right) + \frac{\lambda}{2} \|x\|^2 \right\}$$

ML model represented by d parameters

j*-th training data point on machine *i

The Problem

 # machines

 # training data points on each machine

L2 regularizer

$$\min_{x \in \mathbb{R}^d} \left\{ \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x) \right) + \frac{\lambda}{2} \|x\|^2 \right\}$$

ML model represented by d parameters

j*-th training data point on machine *i

The Problem

machines

training data points on each machine

L2 regularizer

$$\min_{x\in\mathbb{R}^d} \left\{ \left(\frac{1}{n}\sum_{i=1}^{n} \frac{1}{m}\sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x) \right) + \frac{\lambda}{2}\|x\|^2 \right\}$$

ML model represented by d parameters

Loss function

$$\varphi_{ij}(x) : \mathbb{R} \to \mathbb{R}$$
$$|\varphi_{ij}''(s) - \varphi_{ij}''(t)| \le \nu |s-t|$$

j*-th training data point on machine *i

The Problem: Local and Global Functions

$$\min_{x \in \mathbb{R}^d} \left\{ \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x) \right) + \frac{\lambda}{2} \|x\|^2 \right\}$$

The Problem: Local and Global Functions

Local function owned by machine *i*: $f_i(x)$

$$\min_{x\in\mathbb{R}^d}\left\{\left(\frac{1}{n}\sum_{i=1}^{n}\underbrace{\frac{1}{m}\sum_{j=1}^{m}\varphi_{ij}(a_{ij}^\top x)}_{f_i(x)}\right)+\frac{\lambda}{2}\|x\|^2\right\}$$

The Problem: Local and Global Functions

Local function owned by machine *i*: $f_i(x)$

$$\min_{x\in\mathbb{R}^d} \left\{ \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \overbrace{\frac{1}{m}\sum_{j=1}^{m} \varphi_{ij}(a_{ij}^\top x)}^{f_i(x)}\right) + \frac{\lambda}{2}\|x\|^2}_{F(x)} \right\}$$

Global function we want to minimize: $F(x)$

Newton's method

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

Newton's method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \boxed{\nabla^2 f_i(x^k)} + \lambda \mathbf{I}_d\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \boxed{\nabla f_i(x^k)} + \lambda x^k\right)$$

Can be computed by machine *i*

Can be computed by machine *i*

Newton's method

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

Can be computed by machine *i*

Can be computed by machine *i*

- ✓ Local quadratic convergence rate independent of the condition number
- ✗ Expensive $O(d^2)$ communication cost

Newton's method

$$x^{k+1} = x^k - \left(\frac{1}{n}\sum_{i=1}^{n} \underbrace{\nabla^2 f_i(x^k)}_{\text{Can be computed by machine } i} + \lambda \mathbf{I}_d\right)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \underbrace{\nabla f_i(x^k)}_{\text{Can be computed by machine } i} + \lambda x^k\right)$$

✅ Local quadratic convergence rate independent of the condition number

❌ Expensive $O(d^2)$ communication cost

NEWTON-STAR

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

NEWTON-STAR

Hessian at the (unknown!) optimum

$$x^* = \arg\min_x F(x)$$

$$x^{k+1} = x^k - \Big(\underbrace{\frac{1}{n}\sum_{i=1}^{n} \boxed{\nabla^2 f_i(x^*)} + \lambda \mathbf{I}_d}_{\nabla^2 F(x^*)}\Big)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k) + \lambda x^k\right)$$

We assume this is known

NEWTON-STAR

Hessian at the (unknown!) optimum

$$x^* = \arg\min_x F(x)$$

$$\nabla^2 F(x^*)$$

$$x^{k+1} = x^k - \Bigg(\underbrace{\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*) + \lambda \mathbf{I}_d}_{\nabla^2 F(x^*)}\Bigg)^{-1} \left(\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x^k) + \lambda x^k\right)$$

We assume this is known

Can be computed by machine *i*

NEWTON-STAR: Local Quadratic Convergence

$x^* = \arg\min_x F(x)$

$|\varphi''_{ij}(s) - \varphi''_{ij}(t)| \leq \nu |s - t|$

$$\|x^{k+1} - \boxed{x^*}\| \leq \frac{\boxed{\nu}}{2(\mu^* + \lambda)} \left(\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \boxed{\|a_{ij}\|^3} \right) \|x^k - \boxed{x^*}\|^2$$

Training data vectors

$a_{ij} \in \mathbb{R}^d$

NEWTON-STAR: Local Quadratic Convergence

$x^* = \arg\min_x F(x)$

$|\varphi''_{ij}(s) - \varphi''_{ij}(t)| \le \nu |s - t|$

$$\|x^{k+1} - \boxed{x^*}\| \le \frac{\boxed{\nu}}{2(\boxed{\mu^*} + \lambda)} \left(\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \boxed{\|a_{ij}\|^3} \right) \|x^k - \boxed{x^*}\|^{2}$$

$\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*) \succeq \mu^* \mathbf{I}_d$

Training data vectors

$a_{ij} \in \mathbb{R}^d$

NEWTON-STAR: Local Quadratic Convergence

$x^* = \arg\min_x F(x)$

$|\varphi''_{ij}(s) - \varphi''_{ij}(t)| \leq \nu |s-t|$

$$\|x^{k+1} - x^*\| \leq \frac{\nu}{2(\mu^* + \lambda)} \left(\frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \|a_{ij}\|^3 \right) \|x^k - x^*\|^2$$

$\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(x^*) \succeq \mu^* \mathbf{I}_d$

Regularization parameter

$\lambda \geq 0$

Training data vectors

$a_{ij} \in \mathbb{R}^d$

NEWTON-STAR: Summary

$$x^{k+1} = x^k - \left(\nabla^2 F(x^*)\right)^{-1} \nabla F(x^k)$$

- ✅ Local quadratic convergence rate independent of the condition number
- ✅ Cheap *O(d)* communication cost
- ❌ The Hessian at the optimum in unknown

Structure of the Hessian

$$\nabla^2 F(x) = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \varphi''_{ij}(a_{ij}^\top x) a_{ij} a_{ij}^\top \right) + \lambda \mathbf{I}_d$$

Structure of the Hessian

$$\nabla^2 F(x) = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \boxed{\varphi_{ij}''(a_{ij}^\top x)} a_{ij} a_{ij}^\top \right) + \lambda \mathbf{I}_d$$

Assumption 1

$\varphi_{ij} : \mathbb{R} \to \mathbb{R}$ is convex

$(\Rightarrow \varphi_{ij}''(t) \geq 0 \quad \forall t)$

Structure of the Hessian

$$\nabla^2 F(x) = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \boxed{\varphi''_{ij}(a_{ij}^\top x)} a_{ij} a_{ij}^\top \right) + \boxed{\lambda} \mathbf{I}_d$$

Assumption 1

$\varphi_{ij} : \mathbb{R} \to \mathbb{R}$ is convex

$(\Rightarrow \varphi''_{ij}(t) \geq 0 \ \ \forall t)$

Assumption 2

$\lambda > 0$

Structure of the Hessian

Rank-1 matrices formed from the training data vectors

$$\nabla^2 F(x) = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \varphi_{ij}''(a_{ij}^\top x) a_{ij} a_{ij}^\top \right) + \lambda \mathbf{I}_d$$

Assumption 1

$\varphi_{ij} : \mathbb{R} \to \mathbb{R}$ is convex

$(\Rightarrow \varphi_{ij}''(t) \geq 0 \quad \forall t)$

Assumption 2

$\lambda > 0$

NEWTON-LEARN

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

NEWTON-LEARN

Desire: Communication-efficient "approximation" of the Hessian

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

$$\mathbf{H}^k = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} h_{ij}^k a_{ij} a_{ij}^\top}_{\approx \nabla^2 f_i(x^k)} + \lambda \mathbf{I}_d$$

NEWTON-LEARN

Desire: Communication-efficient "approximation" of the Hessian

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

$$\mathbf{H}^k = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} h_{ij}^k a_{ij} a_{ij}^\top + \lambda \mathbf{I}_d}_{\approx \nabla^2 f_i(x^k)}$$

Wish list:

- $h_{ij}^k \to \varphi_{ij}''(a_{ij}^\top x^*)$ as $k \to \infty$

NEWTON-LEARN

Desire: Communication-efficient "approximation" of the Hessian

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

$$\mathbf{H}^k = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} h_{ij}^k a_{ij} a_{ij}^\top}_{\approx \nabla^2 f_i(x^k)} + \lambda \mathbf{I}_d$$

Wish list:

- $h_{ij}^k \to \varphi_{ij}''(a_{ij}^\top x^*)$ as $k \to \infty$
- $h_{i:}^{k+1} - h_{i:}^k \in \mathbb{R}^m$ is sparse $\forall i$
  $h_{i:}^k = (h_{i1}^k, h_{i2}^k, \ldots, h_{im}^k)^\top$

NEWTON-LEARN

Desire: Communication-efficient "approximation" of the Hessian

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

$$\mathbf{H}^k = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\frac{1}{m}\sum_{j=1}^{m} h_{ij}^k a_{ij} a_{ij}^\top}_{\approx \nabla^2 f_i(x^k)} + \lambda \mathbf{I}_d$$

Wish list:

- $h_{ij}^k \to \varphi_{ij}''(a_{ij}^\top x^*)$ as $k \to \infty$
- $h_{i:}^{k+1} - h_{i:}^k \in \mathbb{R}^m$ is sparse $\forall i$
  $h_{i:}^k = (h_{i1}^k, h_{i2}^k, \ldots, h_{im}^k)^\top$
- **Local rate independent of the condition number**

Learning Mechanism in NEWTON-LEARN

$$h_{i:}^{k+1} = \big[\underbrace{h_{i:}^{k} + \eta \mathcal{C}_i^k(\varphi_{i:}''(a_{ij}^\top x^k) - h_{i:}^{k})}\big]_+$$

Compressing the update inspired (by first-order method DIANA)

Learning Mechanism in NEWTON-LEARN

$$h_{i:}^{k+1} = \underbrace{\left[h_{i:}^{k} + \eta \mathcal{C}_i^k\left(\varphi_{i:}''(a_{ij}^\top x^k) - h_{i:}^k\right)\right]}_{\text{Compressing the update inspired (by first-order method DIANA)}}_{+}$$

Projection onto nonnegative orthant

Learning Mechanism in NEWTON-LEARN

Compression operator (e.g., sparsification such as Rand-R)

$$\mathbb{E}\left[\|\mathcal{C}_i^k(h)\|^2\right] \leq (1+\omega)\|h\|^2 \quad \forall\, h \in \mathbb{R}^m$$

$$\mathbb{E}\left[\mathcal{C}_i^k(h)\right] = h \quad \forall\, h \in \mathbb{R}^m$$

$$h_{i:}^{k+1} = \Big[\underbrace{h_{i:}^k + \eta\,\mathcal{C}_i^k\big(\varphi_{i:}''(a_{ij}^\top x^k) - h_{i:}^k\big)}\Big]_+$$

Compressing the update inspired (by first-order method DIANA)

Projection onto nonnegative orthant

Learning Mechanism in NEWTON-LEARN

$$h = \begin{pmatrix} 1 \\ -2 \\ 4 \\ 5 \\ -4 \end{pmatrix} \longrightarrow \mathcal{C}(h) = \frac{5}{2} \begin{pmatrix} 0 \\ -2 \\ 0 \\ 5 \\ 0 \end{pmatrix}$$

Example of Rand-2 Compressor

Compression operator (e.g., sparsification such as Rand-R)

$$\mathbb{E}\left[\|\mathcal{C}_i^k(h)\|^2\right] \leq (1+\omega)\|h\|^2 \quad \forall\, h \in \mathbb{R}^m$$

$$\mathbb{E}\left[\mathcal{C}_i^k(h)\right] = h \quad \forall\, h \in \mathbb{R}^m$$

$$h_{i:}^{k+1} = \Big[\underbrace{h_{i:}^k + \eta\, \mathcal{C}_i^k(\varphi_{i:}''(a_{ij}^\top x^k) - h_{i:}^k)}\Big]_+$$

Compressing the update inspired (by first-order method DIANA)

Projection onto nonnegative orthant

Learning Mechanism in NEWTON-LEARN

$$h = \begin{pmatrix} 1 \\ -2 \\ 4 \\ 5 \\ -4 \end{pmatrix} \rightarrow \mathcal{C}(h) = \frac{5}{2} \begin{pmatrix} 0 \\ -2 \\ 0 \\ 5 \\ 0 \end{pmatrix}$$

Example of Rand-2 Compressor

Compression operator (e.g., sparsification such as Rand-R)

$$\mathbb{E}\left[\|\mathcal{C}_i^k(h)\|^2\right] \leq (1+\omega)\|h\|^2 \quad \forall\, h \in \mathbb{R}^m$$

$$\mathbb{E}\left[\mathcal{C}_i^k(h)\right] = h \quad \forall\, h \in \mathbb{R}^m$$

$$h_{i:}^{k+1} = \Big[\underbrace{h_{i:}^k + \eta \boxed{\mathcal{C}_i^k}(\varphi''_{i:}(a_{ij}^\top x^k) - h_{i:}^k)}\Big]_{\boxed{+}}$$

Rand-R has variance parameter

$$\omega = \frac{m}{r} - 1$$

Compressing the update inspired (by first-order method DIANA)

Projection onto nonnegative orthant

Learning Mechanism in NEWTON-LEARN

Stepsize $0 < \eta \leq \dfrac{1}{\omega + 1}$

Compression operator (e.g., sparsification such as Rand-R)

$$\mathbb{E}\left[\|\mathcal{C}_i^k(h)\|^2\right] \leq (1+\omega)\|h\|^2 \quad \forall\, h \in \mathbb{R}^m$$
$$\mathbb{E}\left[\mathcal{C}_i^k(h)\right] = h \quad \forall\, h \in \mathbb{R}^m$$

$$h_{i:}^{k+1} = \underbrace{\left[h_{i:}^k + \eta\, \mathcal{C}_i^k\left(\varphi_{i:}''(a_{ij}^\top x^k) - h_{i:}^k\right)\right]}_{}{}_{+}$$

Compressing the update inspired (by first-order method DIANA)

Projection onto nonnegative orthant

NEWTON-LEARN: Local Linear Convergence

This is a local result:

$$\left\|x^0 - x^*\right\| \leq \frac{\lambda}{2\sqrt{3}\nu R^3}$$

$$\mathbb{E}\left[\Phi_1^k\right] \leq \left(1 - \min\left\{\frac{5}{8}, \frac{\eta}{2}\right\}\right)^k \Phi_1^0$$

NEWTON-LEARN: Local Linear Convergence

This is a local result:

$$\left\|x^0 - x^*\right\| \leq \frac{\lambda}{2\sqrt{3}\nu R^3}$$

$$\mathbb{E}\left[\Phi_1^k\right] \leq \left(1 - \min\left\{\frac{5}{8}, \frac{\eta}{2}\right\}\right)^k \Phi_1^0$$

Lyapunov function

$$\Phi_1^k := \left\|x^k - x^*\right\|^2 + \frac{1}{3\eta\nu^2 R^2} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \left|h_{ij}^k - \varphi''(a_{ij}^\top x^*)\right|^2$$

$$R := \max_{ij} \|a_{ij}\|$$

NEWTON-LEARN: Local Linear Convergence

This is a local result:

$$\|x^0 - x^*\| \leq \frac{\lambda}{2\sqrt{3}\nu R^3}$$

Rate depends on the compressor only

$$\mathbb{E}\left[\Phi_1^k\right] \leq \left(1 - \min\left\{\frac{5}{8}, \frac{\eta}{2}\right\}\right)^k \Phi_1^0$$

Lyapunov function

$$\Phi_1^k := \left\|x^k - x^*\right\|^2 + \frac{1}{3\eta\nu^2 R^2} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} |h_{ij}^k - \varphi''(a_{ij}^\top x^*)|^2$$

$$R := \max_{ij} \|a_{ij}\|$$

NEWTON-LEARN: Local Linear Convergence

This is a local result:

$$\|x^0 - x^*\| \leq \frac{\lambda}{2\sqrt{3}\nu R^3}$$

Rate depends on the compressor only

$$\mathbb{E}\left[\Phi_1^k\right] \leq \left(1 - \min\left\{\frac{5}{8}, \frac{\eta}{2}\right\}\right)^k \Phi_1^0$$

Lyapunov function

$$\Phi_1^k := \|x^k - x^*\|^2 + \frac{1}{3\eta\nu^2 R^2}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}|h_{ij}^k - \varphi''(a_{ij}^\top x^*)|^2$$

$$R := \max_{ij}\|a_{ij}\|$$

$h_{ij}^k \to \varphi_{ij}''(a_{ij}^\top x^*)$ as $k \to \infty$

We provably learn the Hessian

Further results

Method	Convergence result †	type	rate	Rate independent of the condition number?	Theorem
NEWTON-STAR (NS) (12)	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	2.1
MAX-NEWTON (MN) Algorithm 4	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	D.1
NEWTON-LEARN (NL1) Algorithm 1	$\Phi_1^k \leq \theta_1^k \Phi_1^0$	local	linear	✓	3.2
	$r_{k+1} \leq c\theta_1^k r_k$	local	superlinear	✓	3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$	local	linear	✓	3.5
	$r_{k+1} \leq c\theta_2^k r_k$	local	superlinear	✓	3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$	global	sublinear	✗	4.3
	$\Delta_k \leq c\exp(-k/c)$	global	linear	✗	4.4
	$\Phi_3^k \leq \theta_3^k \Phi_3^0$	local	linear	✓	4.5
	$r_{k+1} \leq c\theta_3^k r_k$	local	superlinear	✓	4.5

Quantities for which we prove convergence: (i) distance to solution $r_k := \left\|x^k - x^*\right\|$; (ii) Lyapunov function $\Phi_q^k := \left\|x^k - x^*\right\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$ for $q = 1,2,3$, where $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$ (see (5)); (iii) Function value suboptimality $\Delta_k := P(x^k) - P(x^*)$

† constant c is possibly different each time it appears in this table. Refer to the precise statements of the theorems for the exact values.

Further results

Method	Convergence			Rate independent of the condition number?	Theorem
	result †	type	rate		
NEWTON-STAR (NS) (12)	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	2.1
MAX-NEWTON (MN) Algorithm 4	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	D.1
NEWTON-LEARN (NL1) Algorithm 1	$\Phi_1^k \leq \theta_1^k \Phi_1^0$	local	linear	✓	3.2
	$r_{k+1} \leq c \theta_1^k r_k$	local	superlinear	✓	3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$	local	linear	✓	3.5
	$r_{k+1} \leq c \theta_2^k r_k$	local	superlinear	✓	3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$	global	sublinear	✗	4.3
	$\Delta_k \leq c \exp(-k/c)$	global	linear	✗	4.4
	$\Phi_3^k \leq \theta_3^k \Phi_3^0$	local	linear	✓	4.5
	$r_{k+1} \leq c \theta_3^k r_k$	local	superlinear	✓	4.5

Quantities for which we prove convergence: (i) distance to solution $r_k := \left\|x^k - x^*\right\|$; (ii) Lyapunov function $\Phi_q^k := \left\|x^k - x^*\right\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$ for $q = 1, 2, 3$, where $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$ (see (5)); (iii) Function value suboptimality $\Delta_k := P(x^k) - P(x^*)$

† constant c is possibly different each time it appears in this table. Refer to the precise statements of the theorems for the exact values.

Further results

Method	Convergence result [†]	Convergence type	Convergence rate	Rate independent of the condition number?	Theorem
NEWTON-STAR (NS) (12)	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	2.1
MAX-NEWTON (MN) Algorithm 4	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	D.1
NEWTON-LEARN (NL1) Algorithm 1	$\Phi_1^k \leq \theta_1^k \Phi_1^0$	local	linear	✓	3.2
	$r_{k+1} \leq c\theta_1^k r_k$	local	superlinear	✓	3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$	local	linear	✓	3.5
	$r_{k+1} \leq c\theta_2^k r_k$	local	superlinear	✓	3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$	global	sublinear	✗	4.3
	$\Delta_k \leq c\exp(-k/c)$	global	linear	✗	4.4
	$\Phi_3^k \leq \theta_3^k \Phi_3^0$	local	linear	✓	4.5
	$r_{k+1} \leq c\theta_3^k r_k$	local	superlinear	✓	4.5

Quantities for which we prove convergence: (i) distance to solution $r_k := \left\|x^k - x^*\right\|$; (ii) Lyapunov function $\Phi_q^k := \left\|x^k - x^*\right\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$ for $q = 1, 2, 3$, where $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$ (see (5)); (iii) Function value suboptimality $\Delta_k := P(x^k) - P(x^*)$

[†] constant c is possibly different each time it appears in this table. Refer to the precise statements of the theorems for the exact values.

Further results

NL2: handles the non-regularized case $\lambda = 0$

Method	Convergence result [†]	type	rate	Rate independent of the condition number?	Theorem
NEWTON-STAR (NS) (12)	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	2.1
MAX-NEWTON (MN) Algorithm 4	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	D.1
NEWTON-LEARN (NL1) Algorithm 1	$\Phi_1^k \leq \theta_1^k \Phi_1^0$	local	linear	✓	3.2
	$r_{k+1} \leq c\theta_1^k r_k$	local	superlinear	✓	3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$	local	linear	✓	3.5
	$r_{k+1} \leq c\theta_2^k r_k$	local	superlinear	✓	3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$	global	sublinear	✗	4.3
	$\Delta_k \leq c\exp(-k/c)$	global	linear	✗	4.4
	$\Phi_3^k \leq \theta_3^k \Phi_3^0$	local	linear	✓	4.5
	$r_{k+1} \leq c\theta_3^k r_k$	local	superlinear	✓	4.5

Quantities for which we prove convergence: (i) distance to solution $r_k := \left\|x^k - x^*\right\|$; (ii) Lyapunov function $\Phi_q^k := \left\|x^k - x^*\right\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$ for $q = 1, 2, 3$, where $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$ (see (5)); (iii) Function value suboptimality $\Delta_k := P(x^k) - P(x^*)$

[†] constant c is possibly different each time it appears in this table. Refer to the precise statements of the theorems for the exact values.

Further results

NL2: handles the non-regularized case $\lambda = 0$

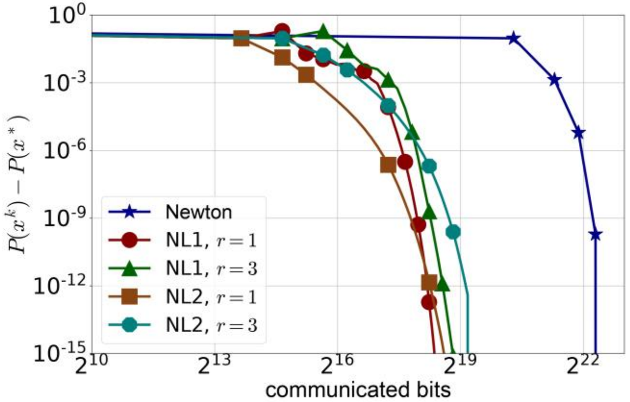
Method	Convergence result [†]	type	rate	Rate independent of the condition number?	Theorem
NEWTON-STAR (NS) (12)	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	2.1
MAX-NEWTON (MN) Algorithm 4	$r_{k+1} \leq c r_k^2$	local	quadratic	✓	D.1
NEWTON-LEARN (NL1) Algorithm 1	$\Phi_1^k \leq \theta_1^k \Phi_1^0$	local	linear	✓	3.2
	$r_{k+1} \leq c\theta_1^k r_k$	local	superlinear	✓	3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$	local	linear	✓	3.5
	$r_{k+1} \leq c\theta_2^k r_k$	local	superlinear	✓	3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$	global	sublinear	✗	4.3
	$\Delta_k \leq c\exp(-k/c)$	global	linear	✗	4.4
	$\Phi_3^k \leq \theta_3^k \Phi_3^0$	local	linear	✓	4.5
	$r_{k+1} \leq c\theta_3^k r_k$	local	superlinear	✓	4.5

Quantities for which we prove convergence: (i) distance to solution $r_k := \left\|x^k - x^*\right\|$; (ii) Lyapunov function $\Phi_q^k := \left\|x^k - x^*\right\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$ for $q = 1, 2, 3$, where $h_{ij}(x^*) = \varphi_{ij}(a_{ij}^\top x^*)$ … tion value suboptimality $\Delta_k := P(x^k) - P(x^*)$

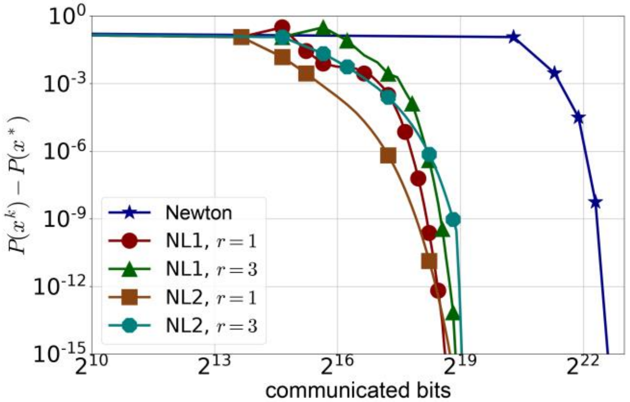
[†] constant c is possibly different each time it appears in this table. Refer to the precise statements of t… exact values.

CNL: Global convergence via cubic regularization

Experiments: comparison with Newton's method



Artificial dataset, $\lambda = 10^{-4}$

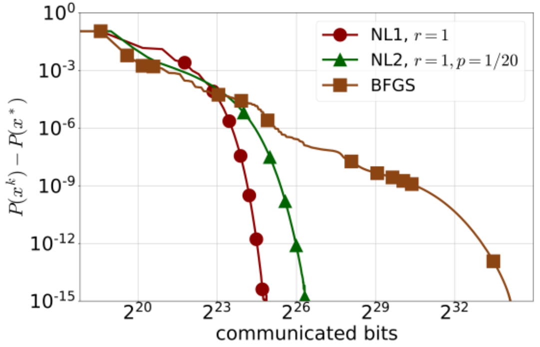


Artificial dataset, $\lambda = 10^{-5}$

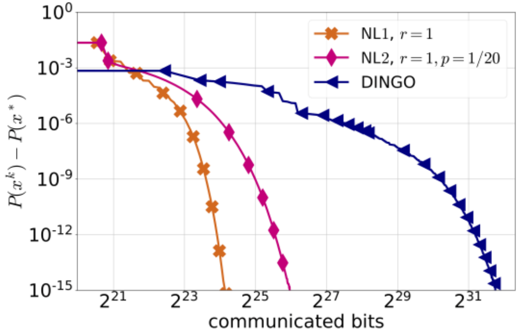
$$\min_{x \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} \log(1 + \exp(-b_{ij} a_{ij}^\top x)) + \frac{\lambda}{2} \|x\|^2 \right\}$$

Logistic regression problem

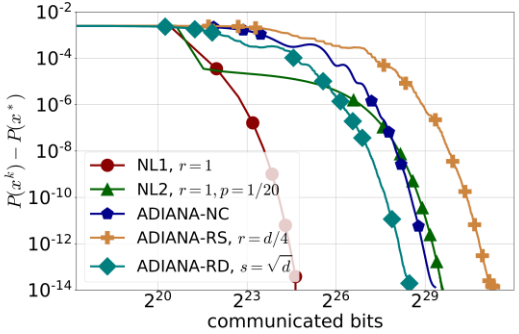
Experiments: comparison with ADIANA, DINGO, BFGS



Comparison of NL with BFGS



Comparison of NL with DINGO



Comparison of NL with ADIANA

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

Logistic regression problem

The End