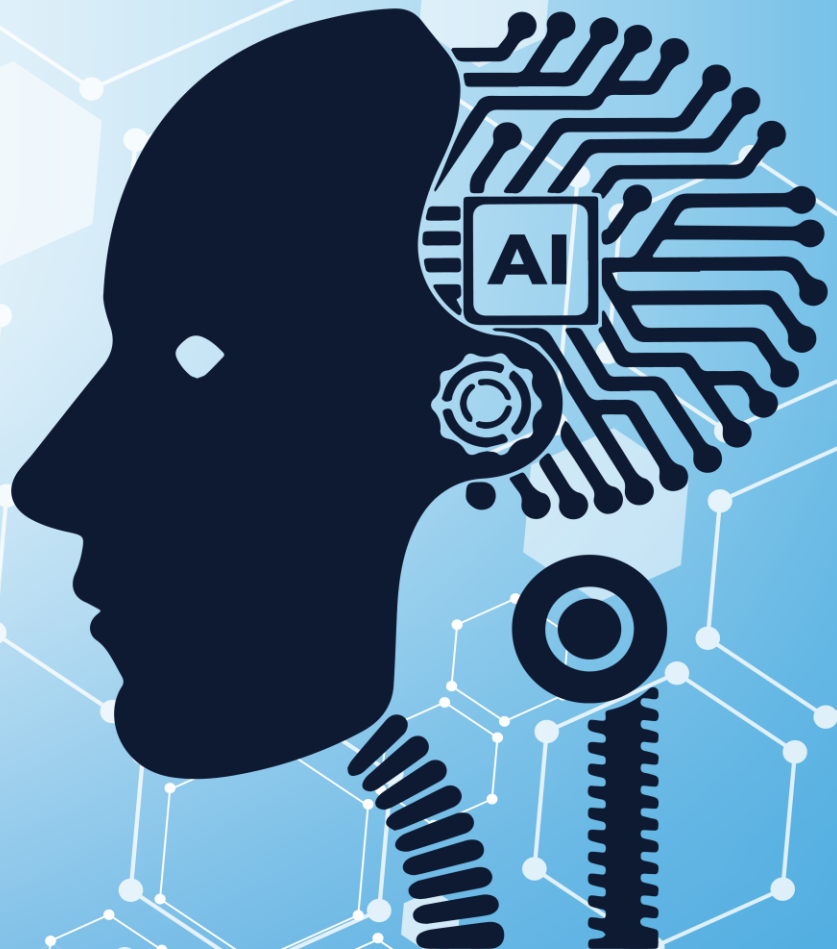


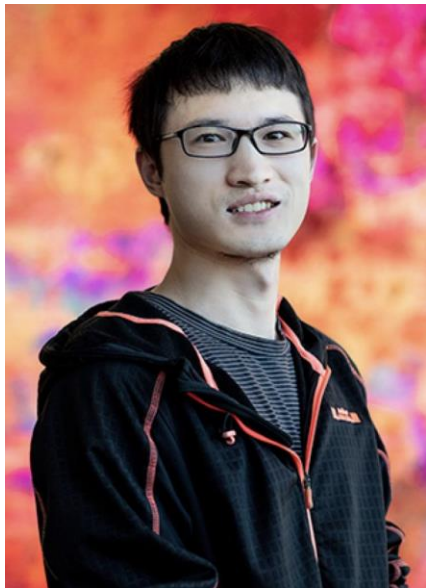
# Distributed Second Order Methods with Fast Rates and Compressed Communication

Maths & AI: MIPT-UGA  
young researchers workshop

Rustem Islamov



# Authors



**Xun Qian**  
**KAUST**



**Peter Richtárik**  
**KAUST**



Rustem Islamov, Xun Qian and Peter Richtárik

**Distributed Second Order Methods with Fast Rates and Compressed Communication**

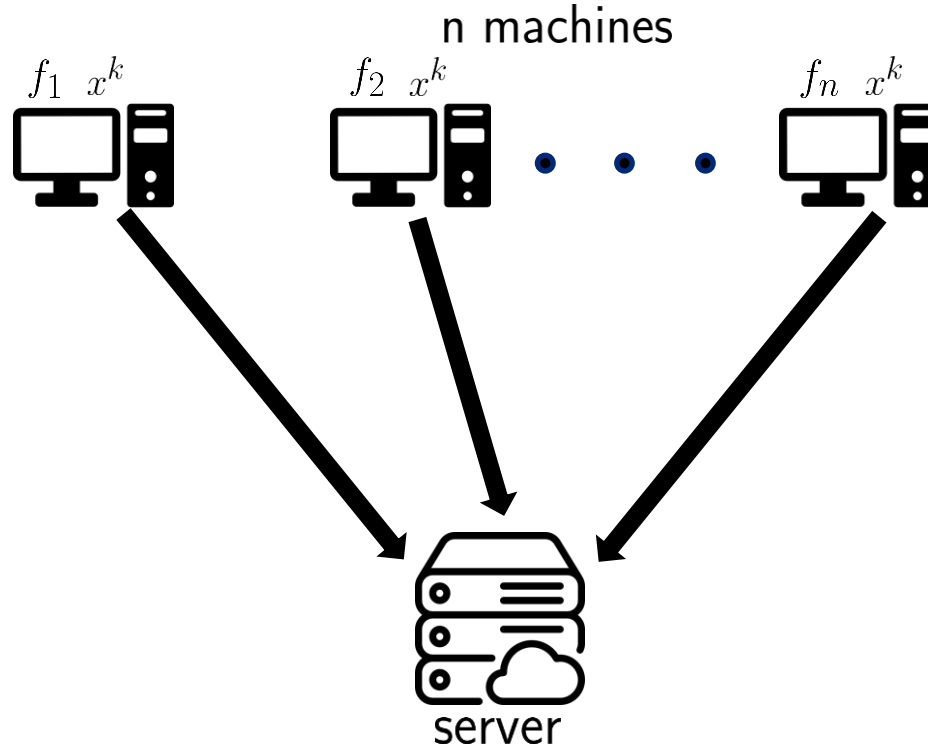
**Accepted to International Conference on Machine Learning 2021**

arXiv:2102.07158, 2021

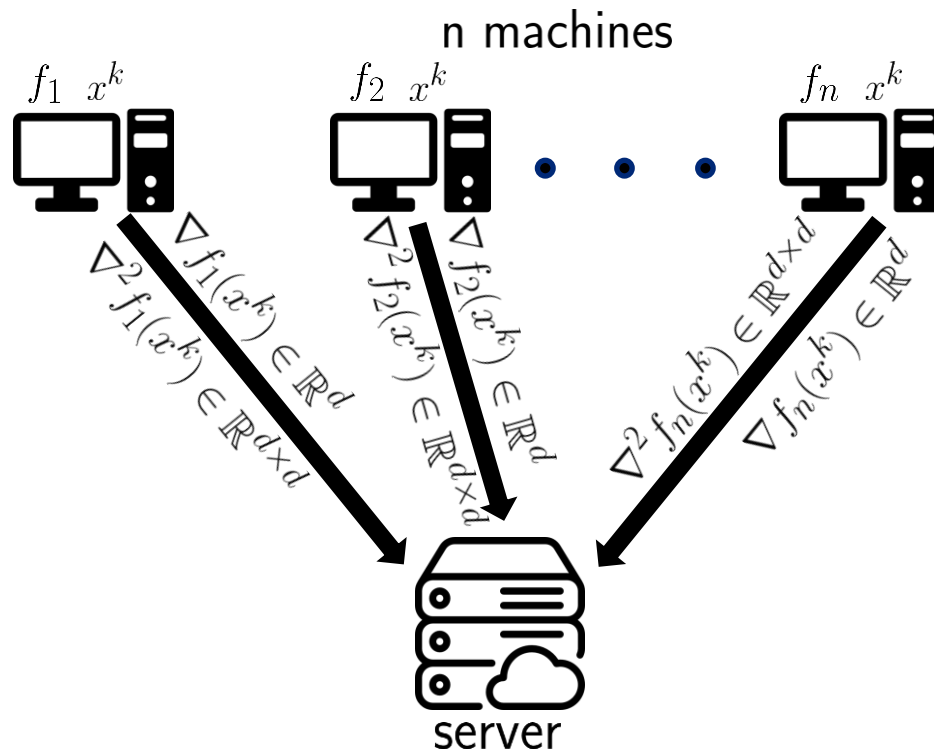
# Outline

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

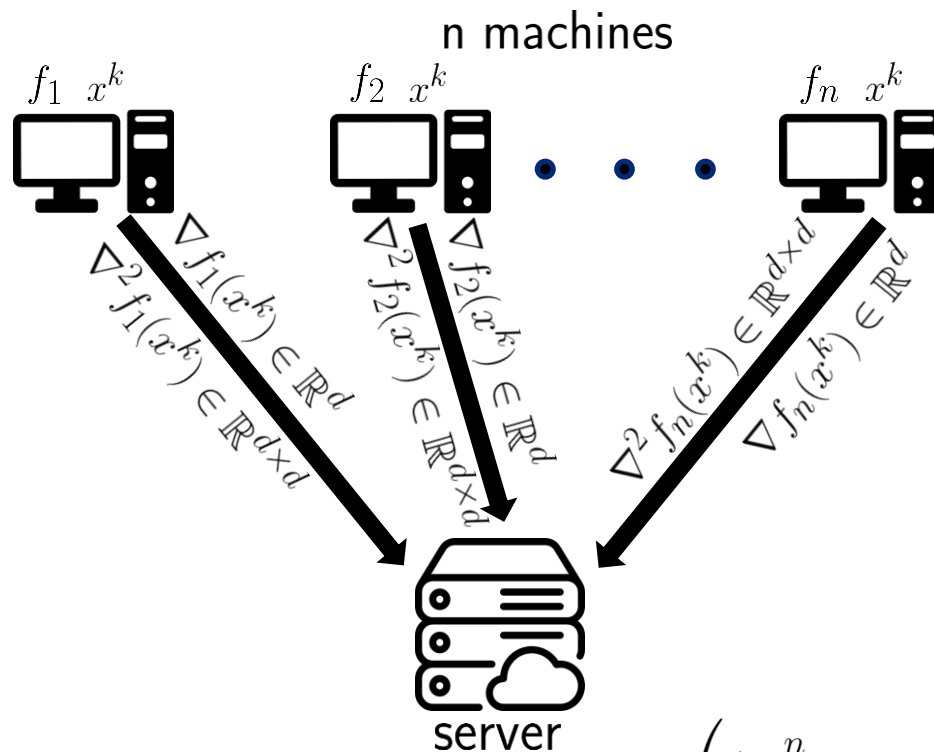
# Server-Client Architecture



# Server-Client Architecture

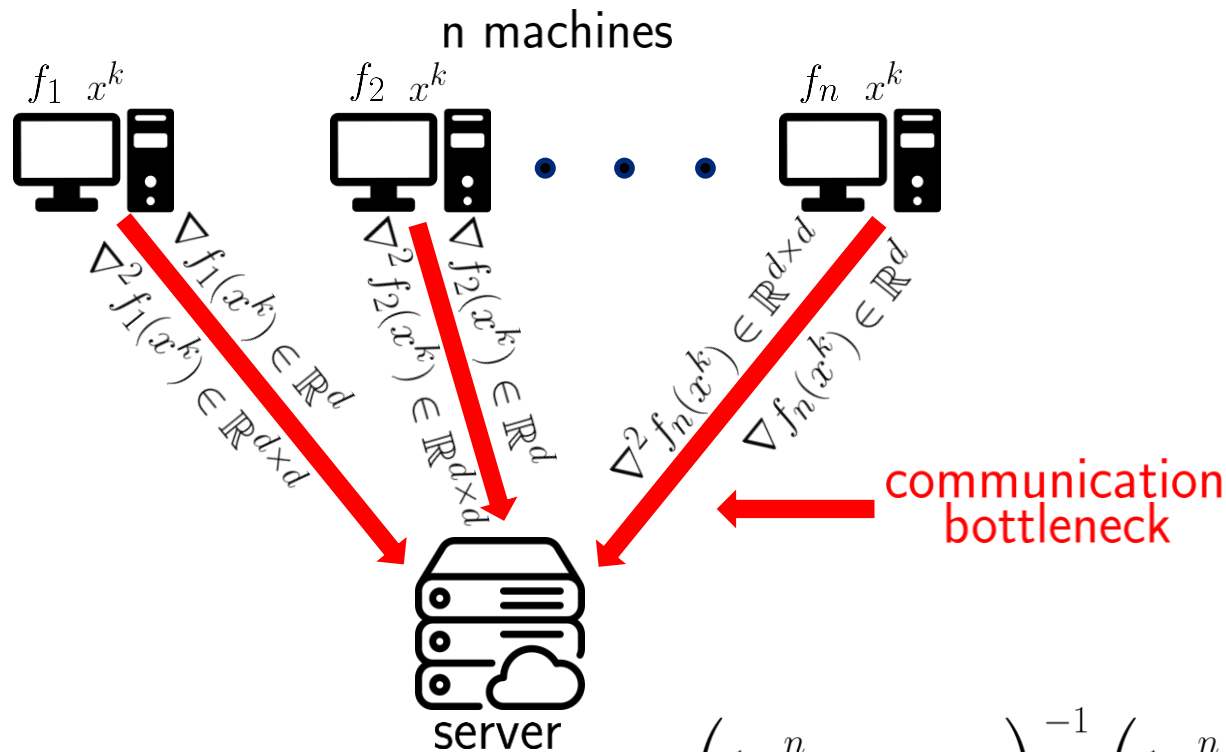


# Server-Client Architecture



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

# 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

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

# 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

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

Examples: QSGD, DCGD,  
DIANA, ADIANA, MARINA

Requires less bits

# 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

The diagram illustrates the problem of minimizing a loss function over a set of parameters  $x \in \mathbb{R}^d$ . The equation is:

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

Callouts explain the variables in the equation:

- # machines**: Points to the variable  $n$  in the denominator of the first sum.
- # training data points on each machine**: Points to the variable  $m$  in the denominator of the second sum.
- ML model represented by  $d$  parameters**: Points to the variable  $x \in \mathbb{R}^d$ .
- $j$ -th training data point on machine  $i$** : Points to the term  $a_{ij}^\top x$  inside the function  $\varphi_{ij}$ .



# The Problem

The diagram illustrates the optimization problem for finding a model  $x \in \mathbb{R}^d$  that minimizes a loss function across  $n$  machines, each with  $m$  training points, plus an L2 regularization term. The variables  $n$ ,  $m$ , and  $x$  are highlighted in colored boxes corresponding to the callouts. The loss function is a sum over machines  $i$  and data points  $j$  of a function  $\varphi_{ij}(a_{ij}^\top x)$ , where  $a_{ij}$  is the  $j$ -th training data point on machine  $i$ . The L2 regularization term is  $\frac{\lambda}{2} \|x\|^2$ .

**# machines**

**# training data points on each machine**

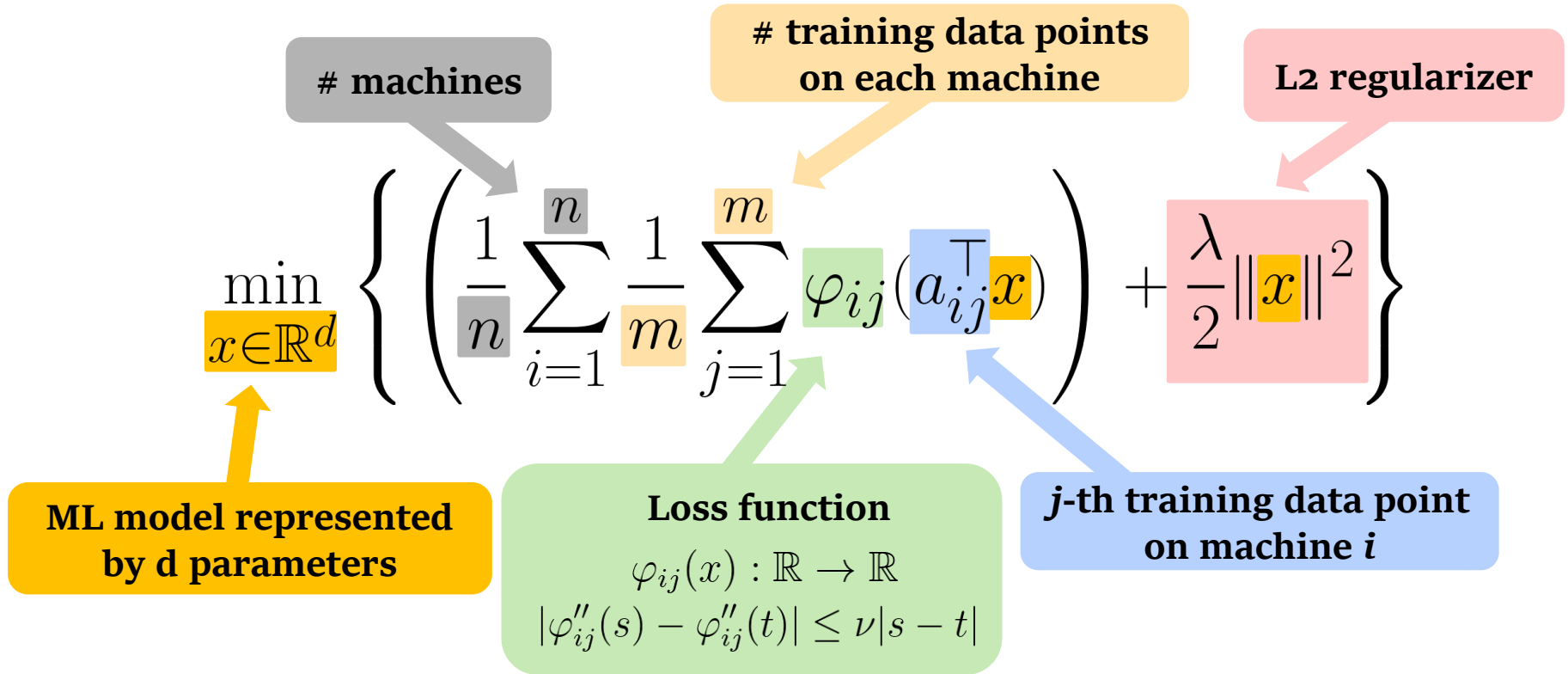
**L2 regularizer**

**ML model represented by  $d$  parameters**

**$j$ -th training data point on machine  $i$**

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

# The Problem



# 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  $i$ :  $f_i(x)$

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

# The Problem: Local and Global Functions

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

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

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

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

Can be computed by machine  $i$

Can be computed by machine  $i$

# Newton's method

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

Diagram illustrating the Newton's method update formula, highlighting the distributed computation of the Hessian and gradient terms:

- The term  $\nabla^2 f_i(x^k)$  (Hessian) is highlighted in a blue box, with an arrow pointing to a box below it stating: **Can be computed by machine  $i$** .
- The term  $\nabla f_i(x^k)$  (Gradient) is highlighted in a blue box, with an arrow pointing to a box below it stating: **Can be computed by machine  $i$** .

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



# Newton's method

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

Diagram illustrating the Newton's method update formula. The formula shows the next iteration  $x^{k+1}$  is calculated by subtracting the product of the inverse of the Hessian matrix (plus a regularization term  $\lambda \mathbf{I}_d$ ) and the gradient (plus a regularization term  $\lambda x^k$ ) from the current point  $x^k$ . The Hessian matrix  $\nabla^2 f_i(x^k)$  and the gradient  $\nabla f_i(x^k)$  are highlighted in blue boxes, with arrows pointing to them from the text "Can be computed by machine  $i$ ".



Local quadratic convergence rate independent of the condition number



Expensive  $O(d^2)$  communication cost

# NEWTON-STAR

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

# NEWTON-STAR

Hessian at the (unknown!) optimum

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

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

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

We assume this is known

# NEWTON-STAR

Hessian at the (unknown!) optimum

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

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

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

We assume this is  
known

Can be computed  
by machine  $i$

# NEWTON-STAR: Local Quadratic Convergence

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

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

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

Training data vectors  
 $a_{ij} \in \mathbb{R}^d$

# NEWTON-STAR: Local Quadratic Convergence

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

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

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

Training data vectors  
 $a_{ij} \in \mathbb{R}^d$

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

# NEWTON-STAR: Local Quadratic Convergence

The diagram illustrates the Newton-STAR convergence theorem. The central inequality is:

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

Annotations and their corresponding mathematical terms:

- Optimal point:**  $x^* = \arg \min_x F(x)$  (orange box) points to  $x^*$  in the inequality.
- Second-order smoothness:**  $|\varphi''_{ij}(s) - \varphi''_{ij}(t)| \leq \nu |s - t|$  (blue box) points to  $\nu$  in the inequality.
- Strong convexity:**  $\frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*) \succeq \mu^* \mathbf{I}_d$  (green box) points to  $\mu^*$  in the denominator.
- Regularization parameter:**  $\lambda \geq 0$  (orange box) points to  $\lambda$  in the denominator.
- Training data vectors:**  $a_{ij} \in \mathbb{R}^d$  (blue box) points to  $\|a_{ij}\|^3$  in the inequality.
- Quadratic convergence:** The exponent 2 on  $\|x^k - x^*\|$  is circled in red.

# 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 is 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

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

**Assumption 1**

$\varphi_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  is convex  
( $\Rightarrow \varphi''_{ij}(t) \geq 0 \quad \forall t$ )

# Structure of the Hessian

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

## Assumption 1

$\varphi_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  is convex  
( $\Rightarrow \varphi''_{ij}(t) \geq 0 \quad \forall t$ )

## 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**

$\varphi_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  is convex  
( $\Rightarrow \varphi''_{ij}(t) \geq 0 \quad \forall t$ )

**Assumption 2**


$\lambda > 0$

# NEWTON-LEARN


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

# NEWTON-LEARN

Desire: Communication-efficient “approximation” of the Hessian




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


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

# NEWTON-LEARN

Desire: Communication-efficient “approximation” of the Hessian



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



## Wish list:

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

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

# NEWTON-LEARN

**Desire: Communication-efficient “approximation” of the Hessian**



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


## Wish list:

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

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



# NEWTON-LEARN

**Desire: Communication-efficient “approximation” of the Hessian**

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

## Wish list:

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

- **Local rate independent of the condition number**

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

# Learning Mechanism in NEWTON-LEARN

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

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

# Learning Mechanism in NEWTON-LEARN

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

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)

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

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

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)

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

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

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

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)

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

Rand-R has  
variance parameter

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

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

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

Projection onto  
nonnegative orthant

# Learning Mechanism in NEWTON-LEARN

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

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

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

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

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

**Projection onto  
nonnegative orthant**

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

# NEWTON-LEARN: Local Linear Convergence

**This is a local result:**

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

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



# NEWTON-LEARN: Local Linear Convergence

This is a local result:

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

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

**Lyapunov function**

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

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

# 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

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

**Lyapunov function**

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

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

# 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

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

**Lyapunov function**

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

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

$$h_{ij}^k \rightarrow \varphi''_{ij}(a_{ij}^\top x^*) \text{ as } k \rightarrow \infty$$

**We provably  
learn the Hessian**

# Further results

| Method                                  | Convergence                         |        |             | Rate independent of the condition number? | Theorem |
|---|-------------------------------------|--------|-------------|---|---------|
|   | result <sup>†</sup>                 | type   | rate        |   |         |
| NEWTON-STAR (NS)<br>(12)                | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | 2.1     |
| MAX-NEWTON (MN)<br>Algorithm 4          | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | D.1     |
| NEWTON-LEARN (NL1)<br>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)<br>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)<br>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  $r_k := \|x^k - x^*\|$ ; (ii) Lyapunov function  $\Phi_q^k := \|x^k - x^*\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$  for  $q = 1, 2, 3$ , where  $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$  (see (5)); (iii) Function value suboptimality  $\Delta_k := P(x^k) - P(x^*)$

<sup>†</sup> 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 <sup>†</sup>                 | type   | rate        |   |         |
| ➡ NEWTON-STAR (NS)<br>(12)              | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | 2.1     |
| MAX-NEWTON (MN)<br>Algorithm 4          | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | D.1     |
| ➡ NEWTON-LEARN (NL1)<br>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)<br>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)<br>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  $r_k := \|x^k - x^*\|$ ; (ii) Lyapunov function  $\Phi_q^k := \|x^k - x^*\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$  for  $q = 1, 2, 3$ , where  $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$  (see (5)); (iii) Function value suboptimality  $\Delta_k := P(x^k) - P(x^*)$

<sup>†</sup> 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 <sup>†</sup>                 | type   | rate        |   |         |
| ➡ NEWTON-STAR (NS)<br>(12)              | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | 2.1     |
| ➡ MAX-NEWTON (MN)<br>Algorithm 4        | $r_{k+1} \leq cr_k^2$               | local  | quadratic   | ✓   | D.1     |
| ➡ NEWTON-LEARN (NL1)<br>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)<br>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)<br>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  $r_k := \|x^k - x^*\|$ ; (ii) Lyapunov function  $\Phi_q^k := \|x^k - x^*\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$  for  $q = 1, 2, 3$ , where  $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$  (see (5)); (iii) Function value suboptimality  $\Delta_k := P(x^k) - P(x^*)$

<sup>†</sup> 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   |        |             | Rate independent of the condition number? | Theorem |
|---|---|--------|-------------|---|---------|
|   | result <sup>†</sup>   | typ    | rate        |   |         |
| ➡ NEWTON-STAR (NS)<br>(12)              | $r_{k+1} \leq cr_k^2$   | local  | quadratic   | ✓   | 2.1     |
| ➡ MAX-NEWTON (MN)<br>Algorithm 4        | $r_{k+1} \leq cr_k^2$   | local  | quadratic   | ✓   | D.1     |
| ➡ NEWTON-LEARN (NL1)<br>Algorithm 1     | $\Phi_1^k \leq \theta_1^k \Phi_1^0$<br>$r_{k+1} \leq c\theta_1^k r_k$ | local  | linear      | ✓   | 3.2     |
|   |   | local  | superlinear | ✓   | 3.2     |
| ➡ NEWTON-LEARN (NL2)<br>Algorithm 2     | $\Phi_2^k \leq \theta_2^k \Phi_2^0$<br>$r_{k+1} \leq c\theta_2^k r_k$ | local  | linear      | ✓   | 3.5     |
|   |   | local  | superlinear | ✓   | 3.5     |
| CUBIC-NEWTON-LEARN (CNL)<br>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  $r_k := \|x^k - x^*\|$ ; (ii) Lyapunov function  $\Phi_q^k := \|x^k - x^*\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$  for  $q = 1, 2, 3$ , where  $h_{ij}(x^*) = \varphi_{ij}''(a_{ij}^\top x^*)$  (see (5)); (iii) Function value suboptimality  $\Delta_k := P(x^k) - P(x^*)$

<sup>†</sup> 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   |        |                       | Rate independent of the condition number? | Theorem    |
|---|---|--------|-----------------------|---|------------|
|   | result <sup>†</sup>   | type   | rate                  |   |            |
| ➡ NEWTON-STAR (NS)<br>(12)                | $r_{k+1} \leq cr_k^2$   | local  | quadratic             | ✓   | 2.1        |
| ➡ MAX-NEWTON (MN)<br>Algorithm 4          | $r_{k+1} \leq cr_k^2$   | local  | quadratic             | ✓   | D.1        |
| ➡ NEWTON-LEARN (NL1)<br>Algorithm 1       | $\Phi_1^k \leq \theta_1^k \Phi_1^0$<br>$r_{k+1} \leq c\theta_1^k r_k$ | local  | linear<br>superlinear | ✓<br>✓                                    | 3.2<br>3.2 |
| ➡ NEWTON-LEARN (NL2)<br>Algorithm 2       | $\Phi_2^k \leq \theta_2^k \Phi_2^0$<br>$r_{k+1} \leq c\theta_2^k r_k$ | local  | linear<br>superlinear | ✓<br>✓                                    | 3.5<br>3.5 |
| ➡ CUBIC-NEWTON-LEARN (CNL)<br>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  $r_k := \|x^k - x^*\|$ ; (ii) Lyapunov function

$\Phi_q^k := \|x^k - x^*\|^2 + c_q \sum_{i=1}^n \sum_{j=1}^m (h_{ij}^k - h_{ij}(x^*))^2$  for  $q = 1, 2, 3$ , where  $h_{ij}(x^*) = \varphi_{ij}(\omega_{ij}^*)$

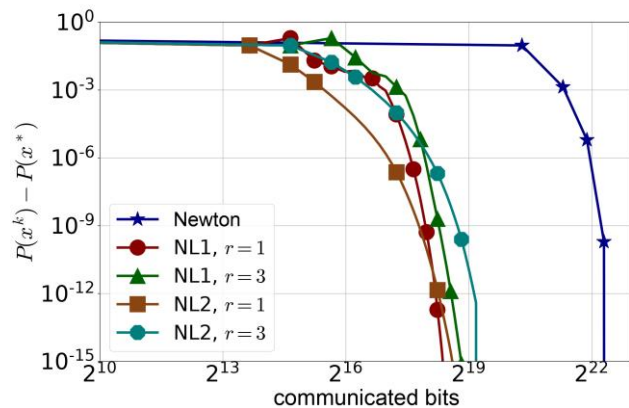
optimization value suboptimality  $\Delta_k := P(x^k) - P(x^*)$

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

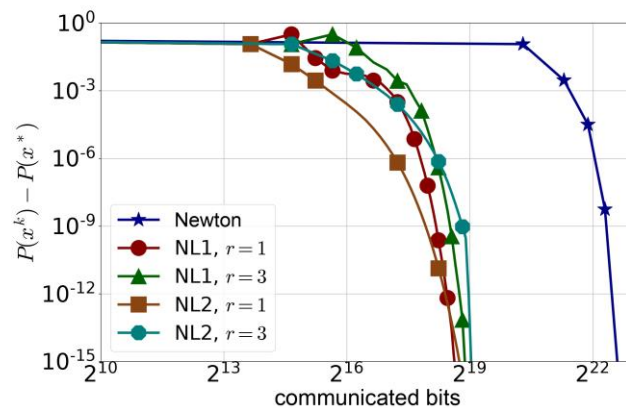
**CNL:** Global convergence via cubic regularization



# Experiments: comparison with Newton's method



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

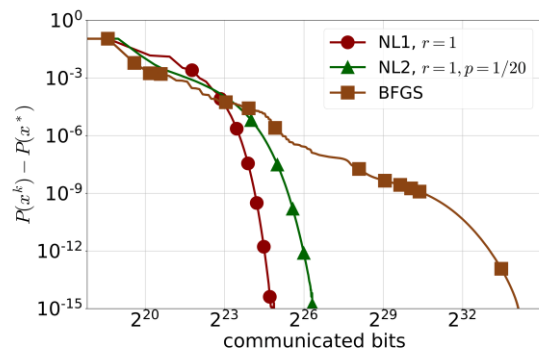


Artificial dataset,  $\lambda = 10^{-5}$

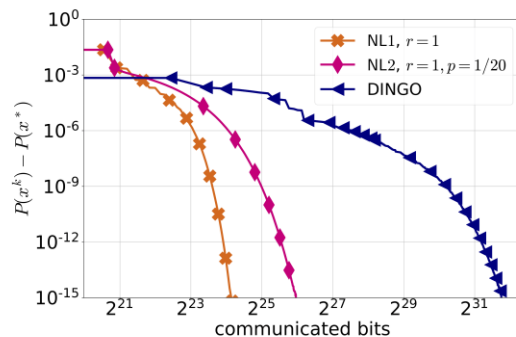
$$\min_{x \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m \log(1 + \exp(-b_{ij} a_{ij}^\top x)) + \frac{\lambda}{2} \|x\|^2 \right\}$$

Logistic regression problem

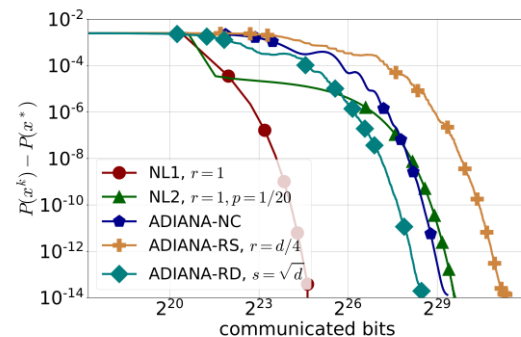
# Experiments: comparison with ADIANA, DINGO, BFGS



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**