Distributed Second Order Methods with Fast Rates and Compressed Communication

KAUST
King Abdullah University of Science and Technology

Peter Richtárik

Optimization without Borders
(Hybrid Event: Online & Sirius University, Sochi, Russia, July 12-16, 2021)

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

Mher Safaryan, Rustem Islamov, Xun Qian and Peter Richtárik
FedNL: Making Newton-type Methods Applicable to Federated Learning
arXiv:2106.02969, 2021

Outline of the Talk

1. Yurii Nesterov’s 65$^{\text{th}}$ Birthday 

2. Introduction
3. NEWTON
4. NEWTON-STAR
5. NEWTON-LEARN
6. Further Results
7. Experiments
8. On Diana and Friends
9. FedNL

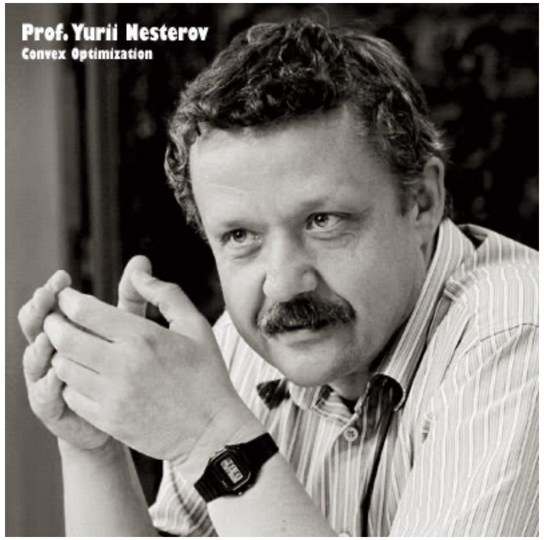
Mher Safaryan, Rustem Islamov, Xun Qian and Peter Richtárik
FedNL: Making Newton-type Methods Applicable to Federated Learning
arXiv:2106.02969, 2021

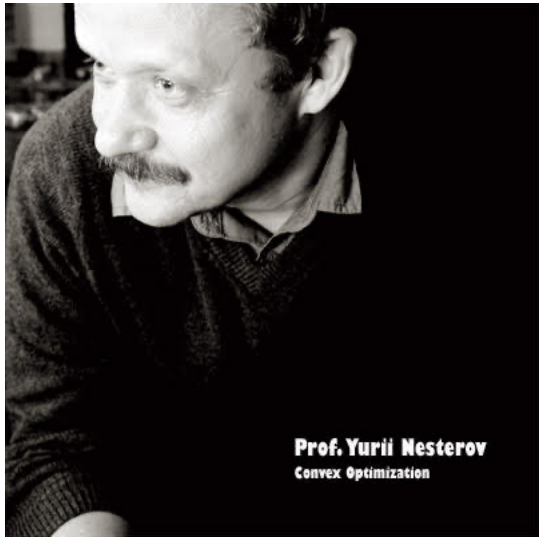
1. Yurii Nesterov’s 65th Birthday

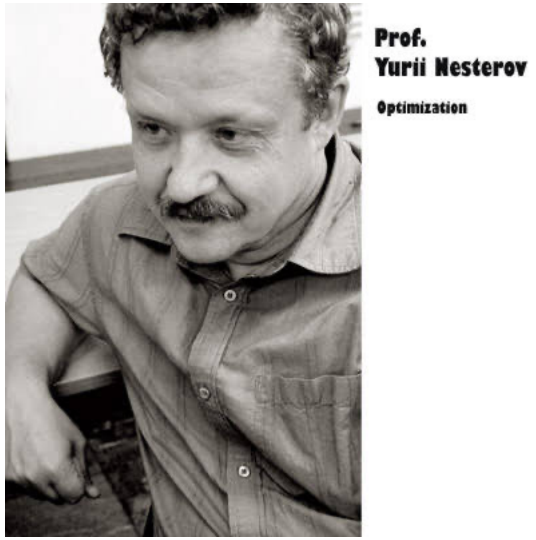
Bruxelles, 2008

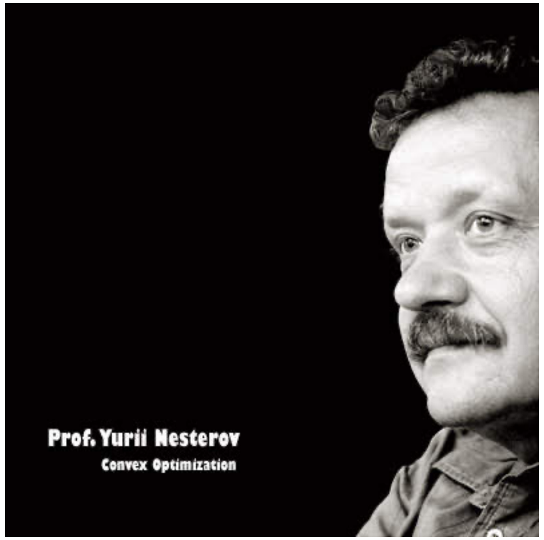

Leuven, 2008

Matagne-la-Petite, 2008









Matagne-la-Petite, 2008

FoCM’11

Optimization Workshop

Conference on the FOUNDATIONS of COMPUTATIONAL MATHEMATICS

Budapest ● July 4-6, 2011

Louvain-la-Neuve, 2011

Edinburgh, 2012


Edinburgh, 2013

Les Houches, 2016

Interacting with Yurii Nesterov was the single most significant and defining moment of my academic life.

Thank you!


You're
Not
65
You're 21
With 44 Years
Experience


YOU ARE TURNING
65
SORRY, YOUR BIRTHDAY
GIFT IS UNDER QUARANTINE

2. Introduction

King Abdullah University of Science and Technology

Optimization and Machine Learning Lab

Openings: research scientists, postdocs, PhD students, MS students, and interns

Research Scientists
Laurent Condat (from Grenoble)
Zhize Li (from Tsinghua)

Postdocs
Mher Safaryan (from Yerevan)
Adil Salim (from Télécom Paris)
Xun Qian (from Hong Kong)

PhD Students
Konstantin Mishchenko (from ENS Paris-Saclay)
Alibek Sailanbayev (from MIPT)
Samuel Horváth (from Comenius)
Elnur Gasanov (from MIPT)
Dmitry Kovalev (from MIPT)
Konstantin Burlachenko (from Huawei)
Slavomír Hanzely (from Comenius)
Lukang Sun (from Nanjing)

MS Students
Egor Shulgin (from MIPT)
Grigory Malinovsky (from MIPT)
Igor Sokolov (from MIPT)

Research Interns
Ilyas Fatkhullin (from Munich)
Rustem Islamov (from MIPT)
Bokun Wang (from UC Davis)

lab composition as of April 2021

Rustem Islamov Home Education Publications Posts Contact CV

Rustem Islamov

Bachelor student

MIPT

CV

Bio

I am a fourth year Bachelor student at Moscow Institute of Physics and Technology. I am interested in Optimization and its applications to Machine Learning. Currently I am working under supervision of Peter Richtárik.

Besides, I am a big fan of football and basketball.

Computer skills

- Operating systems: Microsoft Windows, Linux
- Programming languages: Python, LaTeX, C, C++

Interests

- Optimization
- Machine Learning
- Federated learning

https://rustem-islamov.github.io

Academic Home Publications Contact

Xun Qian

Postdoc

KAUST

Biography

Xun Qian is a postdoc fellow under the supervision of Prof. Peter Richtarik in CEMSE, King Abdullah University of Science and Technology (KAUST). He obtained his PhD degree under the supervision of Prof. Liao Li-Zhi in Department of Mathematics, Hong Kong Baptist University (HKBU) in 2017.

Interests

- Stochastic Methods and Algorithms
- Machine Learning
- Interior Point Methods

Education

- PhD in Mathematics, 2017
  Hong Kong Baptist University
- BSc in Mathematics, 2013
  Huazhong University of Science and Technology

https://qianxunk.github.io

Distributed Second Order Methods with Fast Rates and Compressed Communication

Rustem Islamov* Xun Qian† Peter Richtárik‡

February 13, 2021

Abstract

We develop several new communication-efficient second-order methods for distributed optimization. Our first method, NEWTON-STAR, is a variant of Newton's method from which it inherits its fast local quadratic rate. However, unlike Newton's method, NEWTON-STAR enjoys the same per iteration communication cost as gradient descent. While this method is impractical as it relies on the use of certain unknown parameters characterizing the Hessian of the objective function at the optimum, it serves as the starting point which enables us design practical variants thereof with strong theoretical guarantees. In particular, we design a stochastic sparsification strategy for learning the unknown parameters in an iterative fashion in a communication efficient manner. Applying this strategy to NEWTON-STAR leads to our next method, NEWTON-LEARN, for which we prove local linear and superlinear rates independent of the condition number. When applicable, this method can have dramatically superior convergence behavior when compared to state-of-the-art methods. Finally, we develop a globalization strategy using cubic regularization which leads to our next method, CUBIC-NEWTON-LEARN, for which we prove global sublinear and linear convergence rates, and a fast superlinear rate. Our results are supported with experimental results on real datasets, and show several orders of magnitude improvement on baseline and state-of-the-art methods in terms of communication complexity.

Contents

*King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia, and Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, Russia. This research was conducted while this author was an intern at KAUST and an undergraduate student at MIPT.
†King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.
‡King Abdullah University of Science and Technology, Thuwal, Saudi Arabia.

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

Distributed Second Order Methods with Fast Rates and Compressed Communication

Rustem Islamov[1, 2] Xun Qian[1] Peter Richtárik[1]

[1]KAUST [2]MIPT

The Problem

$$\min_{x\in\mathbb{R}^d}\left[P(x) := f(x) + \frac{\lambda}{2}\|x\|^2\right]. \quad (1)$$

Function f is convex, and has an "average of averages" structure:

$$f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad f_i(x) := \frac{1}{m}\sum_{j=1}^{m} f_{ij}(x), \quad (2)$$

and $\lambda \geq 0$ is a regularization parameter. Each f_{ij} is a function of the form: $f_{ij}(x) := \varphi_{ij}(a_{ij}^\top x)$. The Hessian of f_{ij} at point x is

$$\mathbf{H}_{ij}(x) := h_{ij}(x)a_{ij}a_{ij}^\top, \quad h_{ij}(x) := \varphi''_{ij}(a_{ij}^\top x). \quad (3)$$

The Hessian $\mathbf{H}_i(x)$ of local functions $f_i(x)$ and the Hessian $\mathbf{H}(x)$ of f can be represented as linear combination of one-rank matrices.

Assumptions

We assume that Problem (1) has at least one optimal solution x^*. For all i and j, φ_{ij} is γ-smooth, twice differentiable, and its second derivative φ''_{ij} is ν-Lipschitz continuous.

Main goal

Our goal is to develop a communication efficient Newton-type method for distributed optimization.

Naive distributed implementation of Newton's method

Newton's step: $x^{k+1} \overset{(1)}{=} x^k - \left(\mathbf{H}(x^k) + \lambda\mathbf{I}\right)^{-1}\nabla P(x^k)$.

Each node: computes the local Hessian $\mathbf{H}_i(x^k)$ and gradient $\nabla f_i(x^k)$, then sends them to the server.

Server: averages the local Hessians and gradients to produce $\mathbf{H}(x^k)$ and $\nabla f(x^k)$, respectively, adds $\lambda\mathbf{I}$ to $\mathbf{H}(x^k)$ and λx^k to $\nabla f(x^k)$, then performs Newton step. Next, it sends x^{k+1} back to the nodes.

Pros:
- Fast local quadratic convergence rate
- Rate is independent on the condition number

Cons:
- Requires $\mathcal{O}(d^2)$ floats to be communicated by each worker to the server, where d is typically very large

NEWTON-STAR (NS)

Assume that the server has access to coefficients $h_{ij}(x^*)$ for all i and j, i.e access to the Hessian $\mathbf{H}(x^*)$.

Step of NEWTON-STAR: $x^{k+1} = x^k - (\mathbf{H}(x^*) + \lambda\mathbf{I})^{-1}\nabla P(x^k)$.

Theorem 1 (Convergence of NS)

Assume that $\mathbf{H}(x^*) \succeq \mu^*\mathbf{I}$ for some $\mu^* \geq 0$ and that $\mu^* + \lambda > 0$. Then for any starting point $x^0 \in \mathbb{R}^d$, the iterates of NEWTON-STAR satisfy the following inequality:

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

Pros:
- Fast local quadratic convergence rate
- Rate is **independent on the condition number**
- Communication cost is $\mathcal{O}(d)$ per-iteration

Cons:
- Cannot be implemented in practice

NEWTON-LEARN

How to address the communication bottleneck?
- Compressed communication
- Taking advantage of the structure of the problem

In NEWTON-LEARN we maintain a sequence of vectors

$$h_i^k = (h_{i1}^k, \dots, h_{im}^k) \in \mathbb{R}^m, \quad (4)$$

for all $i = 1, \dots, n$ throughout the iterations $k \geq 0$, with the goal of learning the values $h_{ij}(x^*)$ for all i, j:

$$h_{ij}(x^k) \to h_{ij}(x^*) \quad \text{as} \quad k \to +\infty. \quad (5)$$

Using $h_{ij}^k \approx h_{ij}(x^*)$, we can estimate the Hessian $\mathbf{H}(x^*)$ via

$$\mathbf{H}(x^*) \approx \mathbf{H}^k := \frac{1}{n}\sum_{i=1}^{n}\mathbf{H}_i^k, \quad \mathbf{H}_i^k := \frac{1}{m}\sum_{j=1}^{m} h_{ij}^k a_{ij}a_{ij}^\top. \quad (6)$$

Compressed learning

Compression operator: A randomized map $\mathcal{C} : \mathbb{R}^m \to \mathbb{R}^m$ is a *compression operator (compressor)* if there exists a constant $\omega \geq 0$ such that for all $x \in \mathbb{R}^m$

$$\mathbb{E}\left[\mathcal{C}(x)\right] = x, \quad \mathbb{E}\left[\|\mathcal{C}(x)\|^2\right] \leq (\omega+1)\|x\|^2. \quad (7)$$

Random sparsification (random-r) [1]: Compressor defined as

$$\mathcal{C}(x) := \frac{m}{r} \cdot \xi \circ x, \quad (8)$$

where $\xi \in \mathbb{R}^m$ is a random vector distributed uniformly at random on the discrete set $\{y \in \{0,1\}^m : \|y\|_0 = r\}$. The variance parameter associated with this compressor is $\omega = \frac{m}{r} - 1$.

NEWTON-LEARN: NL1

Assumption: We assume that each $\varphi_{ij}(x)$ is convex, and $\lambda > 0$.

Learning the coefficients: the idea

We design a learning rule for vectors h_i^k via the **DIANA trick** [2] :

$$h_i^{k+1} = \left[h_i^k + \eta\mathcal{C}_i^k\left(h_i(x^k) - h_i^k\right)\right]_+, \quad (9)$$

where $\eta > 0$ is a learning rate, and $\mathcal{C}_i^k$ is a freshly sampled compressor by node i at iteration k.

Main properties:
- $h_{ij}^k \geq 0$ for all i, j
- update is sparse: $\|h_i^{k+1} - h_i^k\|_0 \leq s$, where $s = \mathcal{O}(1)$
- $\mathbf{H}^k \succeq \mathbf{0}$

Each node: Computes update $h_i^{k+1} = \left[h_i^k + \eta\mathcal{C}_i^k\left(h_i(x^k) - h_i^k\right)\right]_+$ and gradient $\nabla f_i(x^k)$. Then the node broadcasts the gradient, update $h_i^{k+1} - h_i^k$ and data points a_{ij} for which $h_{ij}^{k+1} - h_{ij}^k \neq 0$.

Server: averages the local gradients to produce $\nabla f(x^k)$ and constructs $\mathbf{H}^k$ via (6). Then it performs a Newton-like step:

$$x^{k+1} = x^k - \left(\mathbf{H}^k + \lambda\mathbf{I}\right)^{-1}\left(\nabla f(x^k) + \lambda x^k\right), \quad (10)$$

and finally broadcasts x^{k+1} back to the nodes.

Pros
- Local linear and superlinear rates
- Rates are **independent on the condition number**
- Communication cost $\mathcal{O}(d)$ per iteration

Algorithm 1: NL1: NEWTON-LEARN ($\lambda > 0$ case)

Parameters: learning rate $\eta > 0$
Initialization: $x^0 \in \mathbb{R}^d$; $h_1^0, \dots, h_n^0 \in \mathbb{R}^m_+$;
$\mathbf{H}^0 = \frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m h_{ij}^0 a_{ij}a_{ij}^\top \in \mathbb{R}^{d\times d}$
for $k = 0, 1, \dots$ **do**
  Broadcast x^k to all workers
  for *each node* $i = 1, \dots, n$ **do**
    Compute local gradient $\nabla f_i(x^k)$
    $h_i^{k+1} = [h_i^k + \eta\mathcal{C}_i^k(h_i(x^k) - h_i^k)]_+$ Send $\nabla f_i(x^k)$, $h_i^{k+1} - h_i^k$ and corresponding a_{ij} to server
  end
  $x^{k+1} = x^k - \left(\mathbf{H}^k + \lambda\mathbf{I}\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x^k) + \lambda x^k\right)$
  $\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(h_{ij}^{k+1} - h_{ij}^k)a_{ij}a_{ij}^\top$
end

Convergence theory

The analysis relies on the Lyapunov function

$$\Phi_1^k = \left\|x^k - x^*\right\|^2 + \frac{1}{\eta n m \nu^2 R^2}\mathcal{H}^k, \quad \mathcal{H}^k = \sum_{i=1}^{n}\left\|h_i^k - h_i(x^*)\right\|^2,$$

where $R = \max_{i,j}\|a_{ij}\|$.

Theorem 2 (convergence of NL1)

Theorem 2. Let each φ_{ij} is convex, $\lambda > 0$, and $\eta \leq \frac{1}{\omega+1}$. Assume that $\|x^k - x^*\|^2 \leq \frac{\lambda^2}{12\nu^2R^6}$ for all $k \geq 0$. Then for Algorithm 1 we have the inequalities

$$\mathbb{E}[\Phi_1^k] \leq \theta_1^k\Phi_1^0,$$

$$\mathbb{E}\left[\frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2}\right] \leq \theta_1^k\left(6\eta + \frac{1}{2}\right)\frac{\nu^2R^6}{\lambda^2}\Phi_1^0,$$

where $\theta_1 = 1 - \min\left\{\frac{\eta}{2}, \frac{5}{8}\right\}$, which is independent on the condition number.

Assumption on $\|x^k - x^*\|$ can be relaxed using the following lemma:

Lemma 1

Assume h_{ij}^k is a convex combination of $\{h_{ij}(x^0), \dots, h_{ij}(x^k))\}$ for all i, j and k. Assume $\|x^0 - x^*\|^2 \leq \frac{\lambda}{12\nu^2R^6}$. Then

$$\|x^k - x^*\|^2 \leq \frac{\lambda^2}{12\nu^2R^6} \text{for all } k > 0.$$

It is easy to verify that if we choose $h_{ij}^0 = h_{ij}(x^0)$, use the random sparsification compressor (8) and $\eta \leq \frac{1}{\omega+1}$, then h_{ij}^k is always a convex combination of $\{h_{ij}(x^0), \dots, h_{ij}(x^k)\}$ for $k > 0$.

NEWTON-LEARN: NL2

We additionally develop a modified method (NL2) which handles the case where P is μ-strongly convex, $|h_{ij}^k| \leq \gamma$, and $\lambda \geq 0$.

Pros:
- Local linear and superlinear rates
- Rates are **independent on the condition number**
- $\mathcal{O}(d)$ bits are communicated per iteration

CUBIC-NEWTON-LEARN

We also constructed a method (CNL) with global convergence guarantees using cubic regularization [3].

Pros:
- Local linear and superlinear rates
- Global linear rate in the strongly convex case and global sublinear rate in the convex case
- Rates are **independent on the condition number**
- $\mathcal{O}(d)$ bits are communicated per iteration

Experiments

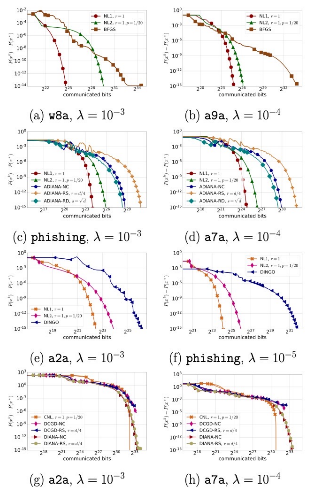


Figure 1:Comparison of NL1, NL2 with (a), (b) BFGS; (c), (d) ADIANA; (e), (f) DINGO in terms of communication complexity. Comparison of CNL with (g), (h) DIANA and DCGD in terms of communication complexity.

References

[1] Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, pages 4447 – 4458, 2018.
[2] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.
[3] Yurii Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1) : 177 – 205, 2006.
[4] Rustem Islamov, Xun Qian, and Peter Richtárik. Distributed Second Order Methods with Fast Rates and Compressed Communication. *arXiv preprint arXiv:2102.07158*, 2021.

Presented on April 9, 2021 at the NSF-TRIPODS Workshop "Communication Efficient Distributed Optimization"

https://richtarik.org/posters/Poster-Newton-Learn.pdf

Embarrassingly Brief Motivation

- Distributed optimization/training is important!
- The **rate of all 1st order methods depends on the condition number**
- Existing 2nd order methods **suffer from at least one of these issues:**
  - **Communication cost** in each communication round is **prohibitively high**
  - Convergence **rate depends on the condition number**

GOAL

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

The Problem

\# machines

\# training data points on each machine

L2 regularizer (optional)

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

ML model represented by *d* parameters / features

Loss function

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

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

j-th training data point on machine *i*

The Problem: Local and Global Functions

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: $F(x)$

3. NEWTON

John Wallis
A treatise of algebra, both historical and practical
Philosophical Transactions of the Royal Society of London, 15(173):1095–1106, 1685

Josepho Raphson
Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, & expedita, ex nova infinitarum serierum methodo, deducta ac demonstrate
Oxford: Richard Davis, 1697

JONATHAN HILL
BOOKSELLER

Analysis Æquationum
UNIVERSALIS,
SEU
Ad ÆQUATIONES ALGEBRAICAS Resolvendas
METHODUS Generalis, & Expedita,
Ex nova Infinitarum Serierum Methodo,
DEDUCTA ac DEMONSTRATA.
Editio Secunda cui accessit Appendix de Infinito Infinitarum Serierum progressu ad Equationum Algebraicarum Radices eliciendas.
Cui etiam Annexum est,
DE
SPATIO REALI
SEU
ENTE INFINITO
CONAMEN Mathematico-Metaphysicum.
Authore JOSEPHO RAPHSON A. M. & Reg. Soc. Socio.
LONDINI,
Typis T. B. prostant venales apud A. & I. Churchill in vico vulgo vocato Pater-noster-Rowe; S. Smith & Ben. Walford; J. Taylor, & T. Bennet in Cœmeterio D. Pauli, Bibliopolas. 1702.

“Raphson’s Method”; Not “Newton’s Method” or,
Maybe, the “Newton–Raphson Method”

RAPHSON, Joseph.

Analysis Aequationum Universalis, seu ad Aequationes Algebraicas resolvendas Methodus generalis, & expedita, ex nova infinitarum serierum methodo, deducta ac demonstrata. Editio secunda cui accessit Appendix de Infinito Infinitarum Serierum progressu ad Equationum Algebraicarum Radices eliciendas. Cui etiam Annexum est; De Spatio reali, seu Ente Infinito Conamen Mathematico-Metaphysicum.

Woodcut diagrams in the text. 3 p.l., 5-55, [9], 95, [1] pp. Small 4to, 18th-cent. calf (rebacked & recornered), red morocco lettering piece on spine. London: Typis TB. for A. & I. Churchill et al., 1702.

Third edition; the first edition appeared in 1690 and the second in 1697. Raphson (d. 1715 or 1716), also wrote the important History of Fluxions (1715) and translated Newton’s Arithmetica Universalis into English (1720). He was a fellow of the Royal Society.

“In 1690, Joseph Raphson…published a tract, Analysis aequationum universalis. His method closely resembles that of Newton. The only difference is this, that Newton derives each successive step, p, q, r, of approach to the root, from a new equation, while Raphson finds it each time by substitution in the original equation…Raphson does not mention Newton; he evidently considered the difference sufficient for his method to be classed independently. To be emphasized is the fact that the process which in modern texts goes by the name of ‘Newton’s method of approximation,’ is really not Newton’s method, but Raphson’s modification of it…It is doubtful, whether this method should be named after Newton alone…Raphson’s version of the process represents what J. Lagrange recognized as an advance on the scheme of Newton… Perhaps the name ‘Newton-Raphson method’ would be a designation more nearly representing the facts of history.”–Cajori, A History of Mathematics, p. 203.

The first edition is very rare. The Appendix appears for the first time in the second edition of 1697 along with the separately paginated second part De Spatio reali.

Fine fresh copy. 19th-century bookplate of P. Duncan.

Price: $4,500.00

Item ID: 3504

ADD TO CART ASK A QUESTION

See all items in Calculus, Mathematics, Newtoniana, Science
See all items by Joseph RAPHSON

Year 1697

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

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

NEWTON

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

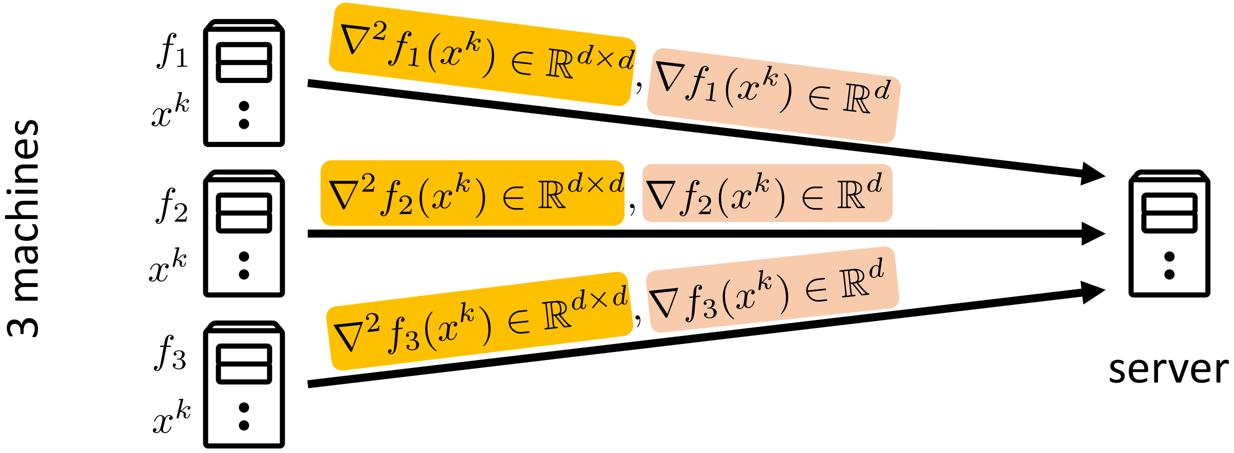
$$\min_{x\in\mathbb{R}^d}\left\{\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\overbrace{\frac{1}{m}\sum_{j=1}^{m}\varphi_{ij}(a_{ij}^\top x)}^{f_i(x)}\right)+\frac{\lambda}{2}\|x\|^2}_{F(x)}\right\}$$

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

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

Can be computed by machine *i*

Can be computed by machine *i*



$$x^{k+1} = x^k - \left(\frac{1}{3}\sum_{i=1}^{3}\nabla^2 f_i(x^k) + \lambda \mathbf{I}_d\right)^{-1}\left(\frac{1}{3}\sum_{i=1}^{3}\nabla f_i(x^k) + \lambda x^k\right)$$

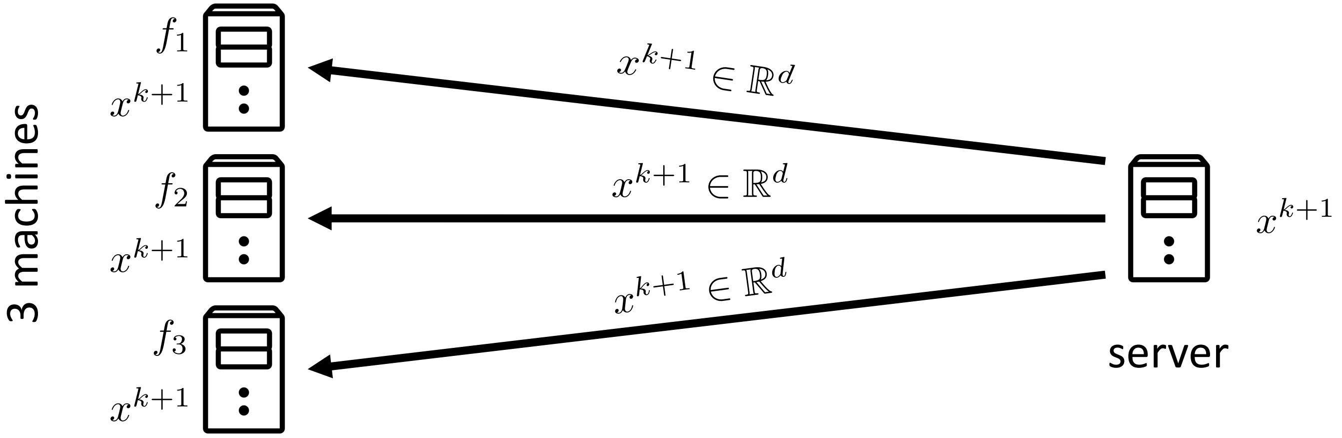
NEWTON

$$\min_{x\in\mathbb{R}^d}\Bigg\{\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\overbrace{\frac{1}{m}\sum_{j=1}^{m}\varphi_{ij}(a_{ij}^\top x)}^{\text{Local function owned by machine } i:\ f_i(x)}\right)+\frac{\lambda}{2}\|x\|^2}_{\text{Global function we want to minimize: } F(x)}\Bigg\}$$

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

Can be computed by machine *i*

Can be computed by machine *i*



Bottleneck of Distributed Implementation of Newton's Method = Communication of *d* x *d* Hessian Matrices!!!

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: Summary

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

Local quadratic convergence independent of the condition number

Expensive $O(d^2)$ worker-master communication

4. NEWTON-STAR
“One Hessian is Enough!”

Hessian at the (unknown!) optimum

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

NEWTON-STAR

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

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

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

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

NEWTON-STAR

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: $F(x)$

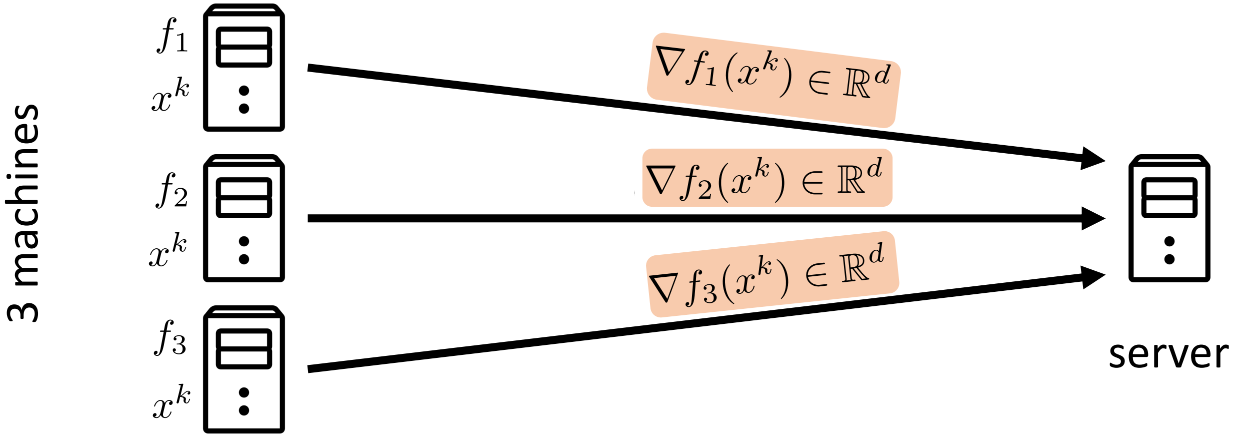
Hessian at the (unknown!) optimum

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

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

We assume this is known!

Can be computed by machine *i*



$$x^{k+1} = x^k - \left(\frac{1}{3}\sum_{i=1}^{3} \nabla^2 f_i(x^\star) + \lambda \mathbf{I}_d \right)^{-1} \left(\frac{1}{3}\sum_{i=1}^{3} \nabla f_i(x^k) + \lambda x^k \right)$$

NEWTON-STAR

Hessian at the (unknown!) optimum

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

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

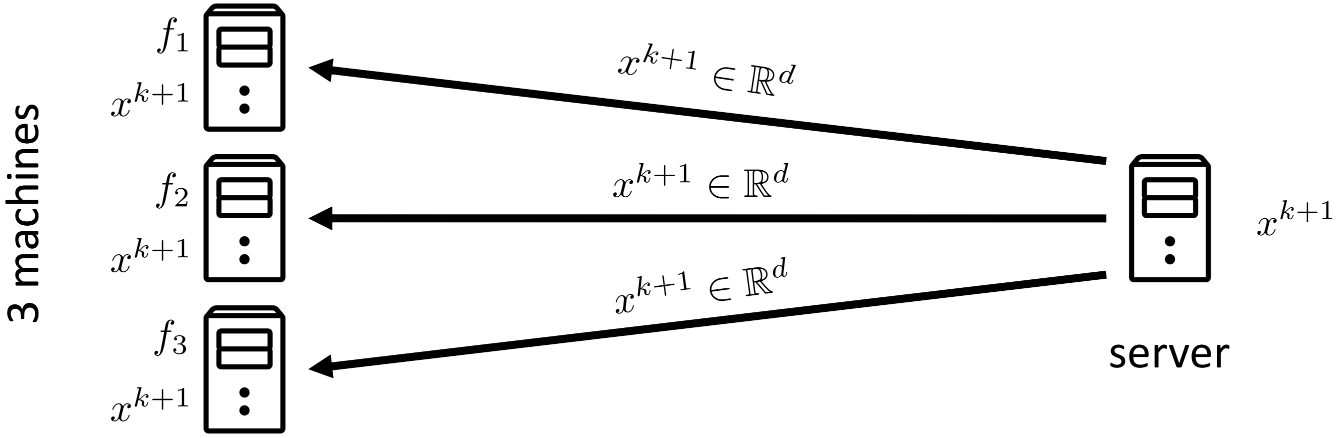
$$\min_{x\in\mathbb{R}^d}\left\{\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\overbrace{\frac{1}{m}\sum_{j=1}^{m}\varphi_{ij}(a_{ij}^\top x)}^{f_i(x)}\right)+\frac{\lambda}{2}\|x\|^2}_{F(x)}\right\}$$

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

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

We assume this is known!

Can be computed by machine *i*



Noo need to communicate any d x d matrices!!!

Same communication cost per iteration as gradient descent!!!

NEWTON-STAR: Local Quadratic Convergence

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: $F(x)$

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

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

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

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

Regularizer parameter

$\lambda \ge 0$

Training data vectors

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

NEWTON-STAR: Summary

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: $F(x)$

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

- Local quadratic convergence independent of the condition number
- Cheap $O(d)$ worker-master communication
- We do not know the Hessian at the optimum!

The New Result From The Previous Slide

5. NEWTON-LEARN
“Let’s Learn the Hessian!”

Structure of the Hessian

$$\min_{x\in\mathbb{R}^d}\left\{\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}\overbrace{\frac{1}{m}\sum_{j=1}^{m}\varphi_{ij}(a_{ij}^\top x)}^{\text{Local function owned by machine } i:\ f_i(x)}\right)+\frac{\lambda}{2}\|x\|^2}_{\text{Global function we want to minimize: } F(x)}\right\}$$

Rank-1 matrices formed from the training data vectors

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

Assumption 1

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

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

Assumption 2

$\lambda > 0$

NEWTON vs NEWTON-STAR

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

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

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

- Local quadratic convergence independent of the condition number
- Expensive $O(d^2)$ worker-master communication

NEWTON-STAR

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

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

- Local quadratic convergence independent of the condition number
- Cheap $O(d)$ worker-master communication
- We do not know the Hessian at the optimum!

We have solved one problem, but introduced a new problem!

NEWTON-LEARN

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: $F(x)$

Desire: Communication-efficient "approximation" of the Hessian

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

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

Wish list:

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

$h_{i:}^{k+1} - h_{i:}^k \in \mathbb{R}^m$ is sparse $\forall i$

$$h_{i:}^k = \begin{pmatrix} h_{i1}^k \\ h_{i2}^k \\ \vdots \\ h_{im}^k\end{pmatrix}\in\mathbb{R}^m$$

local rate independent of condition number

Learning Mechanism in NEWTON-LEARN

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

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

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

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

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

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

Vector of coefficients giving rise to Hessian approximation at machine *i*

$$h_{i:}^k = \begin{pmatrix} h_{i1}^k \\ h_{i2}^k \\ \vdots \\ h_{im}^k \end{pmatrix} \in \mathbb{R}^m \quad \Rightarrow \quad \frac{1}{m}\sum_{j=1}^m h_{ij}^k a_{ij} a_{ij}^\top \approx \nabla^2 f_i(x^k)$$

Projection onto nonnegative orthant

$$z \in \mathbb{R}^m \quad \Rightarrow \quad [z]_+ := \begin{pmatrix} \max\{z_1, 0\} \\ \max\{z_2, 0\} \\ \vdots \\ \max\{z_m, 0\} \end{pmatrix}$$

NEWTON-LEARN: Local Linear Rate Independent of the Condition Number!

This is a local result:

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

Rate depends on the compressor only!

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

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

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

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

Lyapunov function

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

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

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

We provably learn the Hessian!

6. 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	$\Phi_1^k \leq \theta_1^k \Phi_1^0$ $r_{k+1} \leq c\theta_1^k r_k$	local local	linear superlinear	✓ ✓	3.2 3.2
NEWTON-LEARN (NL2) Algorithm 2	$\Phi_2^k \leq \theta_2^k \Phi_2^0$ $r_{k+1} \leq c\theta_2^k r_k$	local local	linear superlinear	✓ ✓	3.5 3.5
CUBIC-NEWTON-LEARN (CNL) Algorithm 3	$\Delta_k \leq \frac{c}{k}$ $\Delta_k \leq c\exp(-k/c)$ $\Phi_3^k \leq \theta_3^k \Phi_3^0$ $r_{k+1} \leq c\theta_3^k r_k$	global global local local	sublinear linear linear superlinear	✗ ✗ ✓ ✓	4.3 4.4 4.5 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 ... exact values.

CNL: Global convergence via cubic regularization (Griewank 1981, Nesterov & Polyak 2006)

7. Experiments

Experimental Setup

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

Table 3: Data sets used in the experiments, and the number of worker nodes n used in each case.

Data set	**# workers n**	**# data points ($= nm$)**	**# features d**
a2a	15	2 265	123
a7a	100	16 100	123
a9a	80	32 560	123
w8a	142	49 700	300
phishing	100	11 000	68
artificial	100	1 000	200

Table 2: Comparison of distributed Newton-type methods. Our methods combine the best of both worlds, and are the only methods we know about which do so: we obtain fast rates independent of the condition number, and allow for $O(d)$ communication per communication round.

SOTA

Method	**Convergence rate**	**Rate independent of the condition number?**	**Communication cost per iteration**	**Network structure**
DANE [Shamir et al., 2014]	Linear	✗	$O(d)$	Centralized
DiSCO [Zhang and Xiao, 2015]	Linear	✗	$O(d)$	Centralized
AIDE [Reddi et al., 2016]	Linear	✗	$O(d)$	Centralized
GIANT [Wang et al., 2018]	Linear	✗	$O(d)$	Centralized
DINGO [Crane and Roosta, 2019]	Linear	✗	$O(d)$	Centralized
DAN [Zhang et al., 2020]	Local quadratic†	✓	$O(nd^2)$	Decentralized
DAN-LA [Zhang et al., 2020]	Superlinear	✓	$O(nd)$	Decentralized
NEWTON-STAR **this work**	Local quadratic	✓	$O(d)$	Centralized
MAX-NEWTON **this work**	Local quadratic	✓	$O(d)$	Centralized
NEWTON-LEARN **this work**	Local superlinear	✓	$O(d)$	Centralized
CUBIC-NEWTON-LEARN **this work**	Superlinear	✓	$O(d)$	Centralized

† DAN converges globally, but the quadratic rate is introduced only after $O(L_2/\mu^2)$ steps, where L_2 is the Lipschitz constant of the Hessian of P, and μ is the strong convexity parameter of P. This is a property it inherits from the recent method of Polyak [Polyak and Tremba, 2019] this method is based on.

NL1 & NL2:
The Effect of Compression

NL1 & NL2: The Effect of Compression

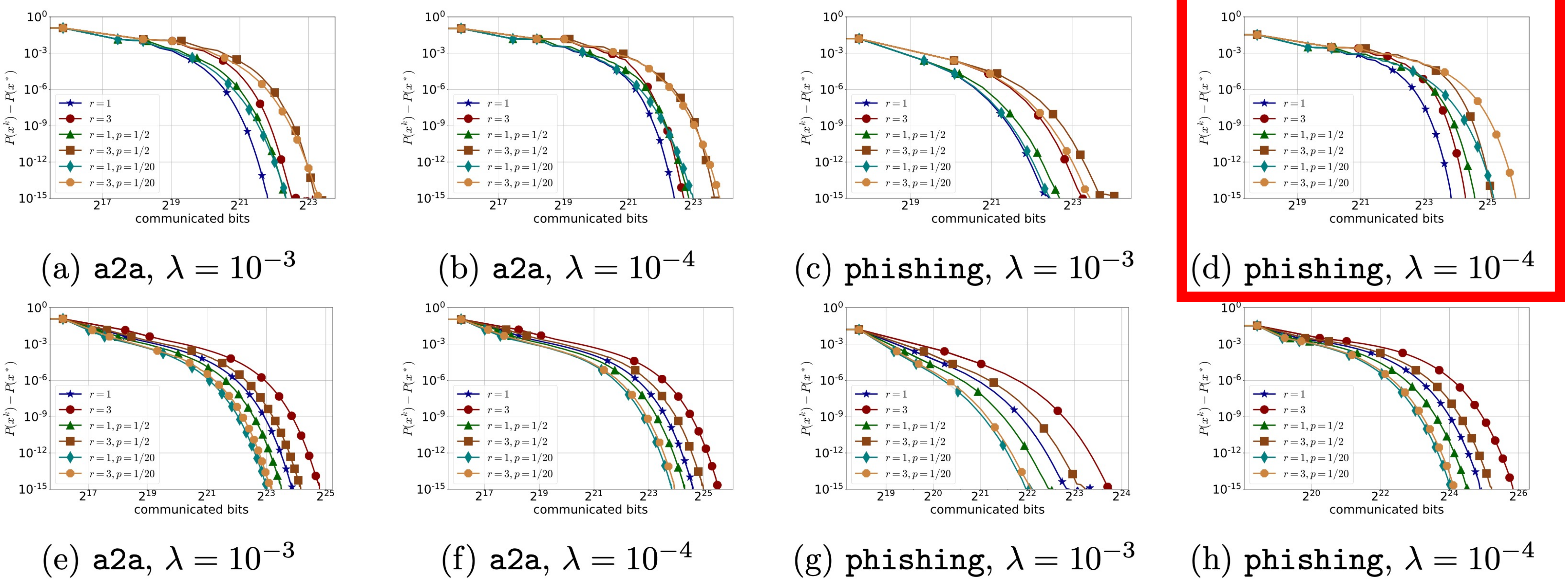
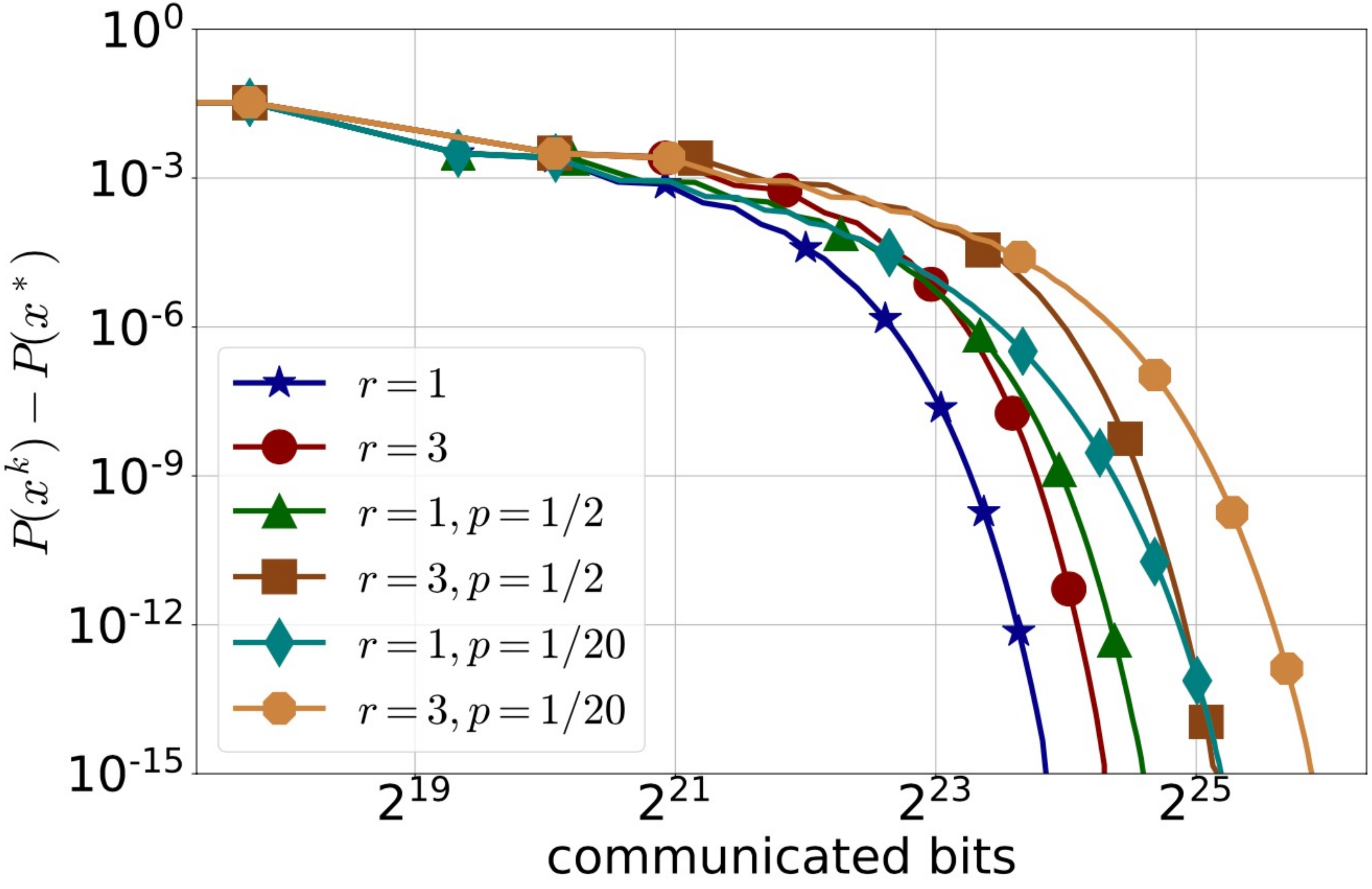


Figure 1: Performance of NL1 (first row) and NL2 (second row) across a few values of r defining the random-r compressor, and a few values of p defining the induced Bernoulli compressor $\mathcal{C}_p$.



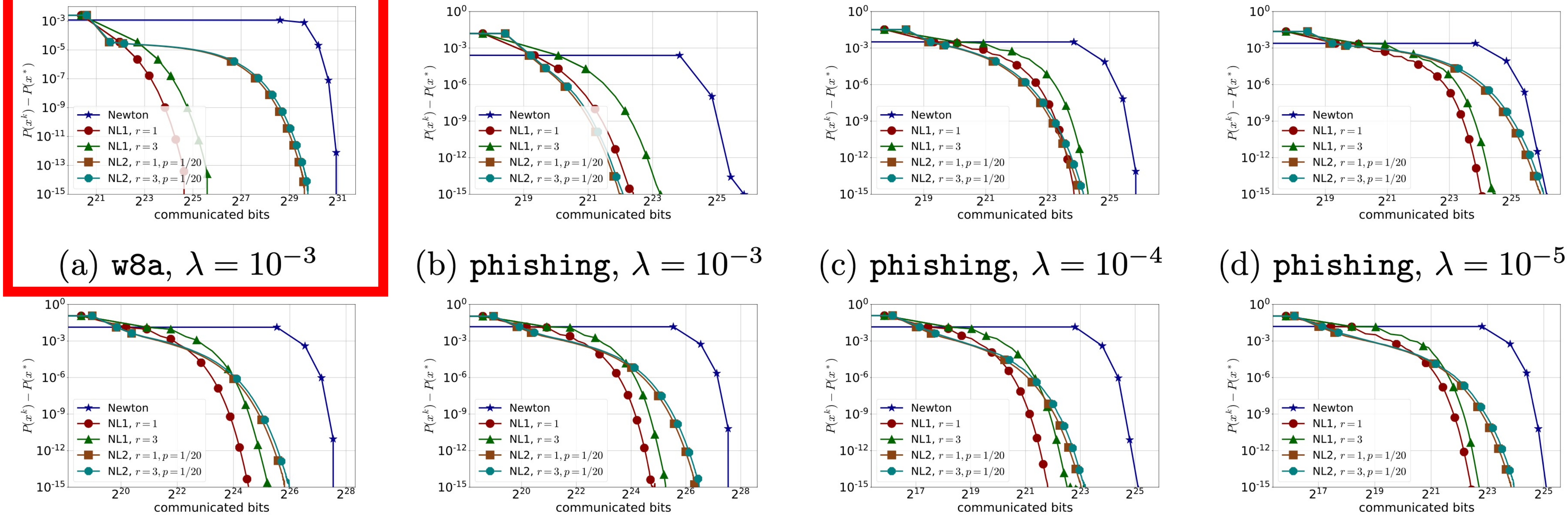
(d) phishing, $\lambda = 10^{-4}$

NL1 & NL2

vs

Newton

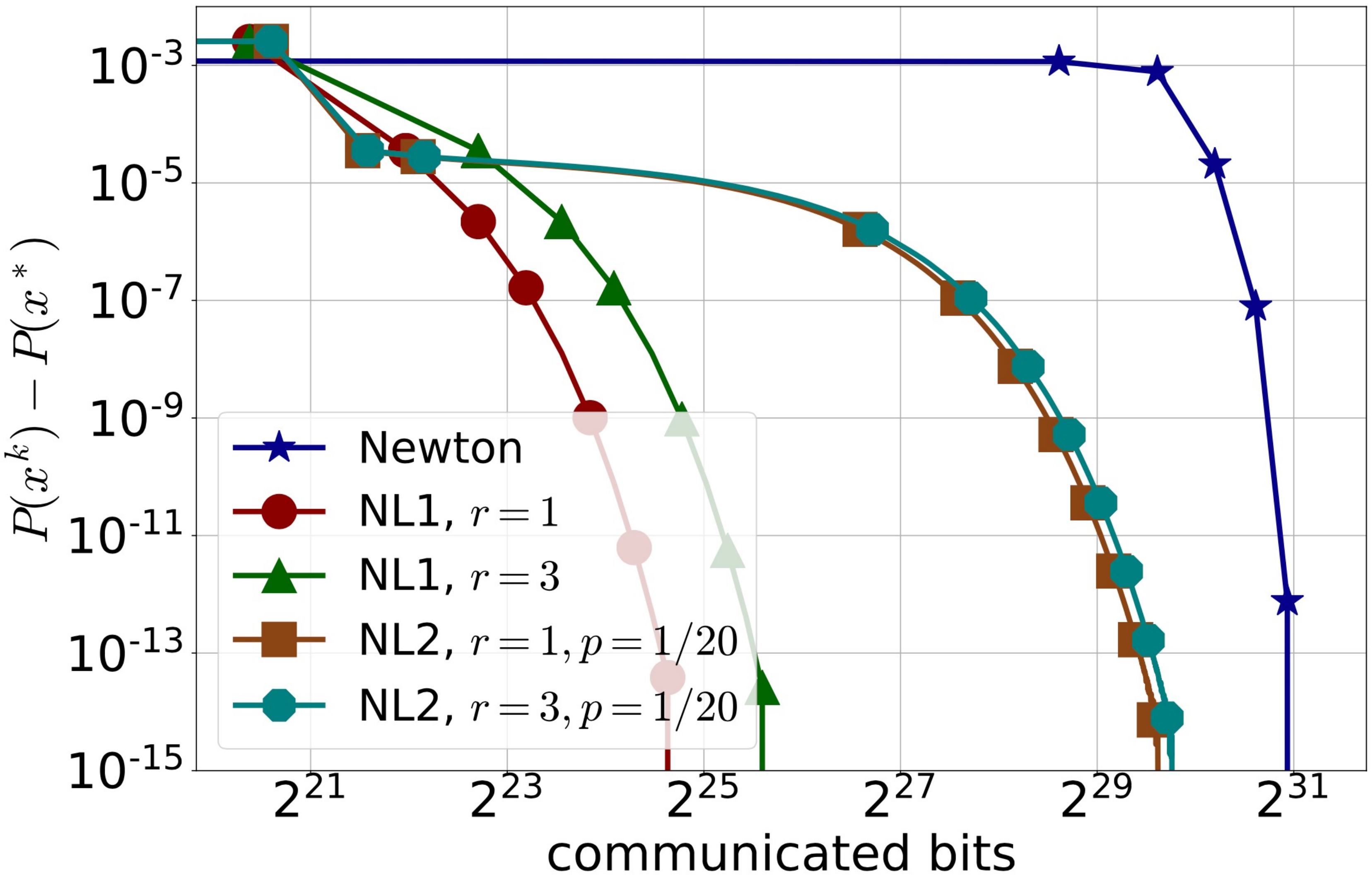
NL1 & NL2 vs Newton



(a) w8a, $\lambda = 10^{-3}$ (b) phishing, $\lambda = 10^{-3}$ (c) phishing, $\lambda = 10^{-4}$ (d) phishing, $\lambda = 10^{-5}$

(e) a7a, $\lambda = 10^{-3}$ (f) a7a, $\lambda = 10^{-3}$ (g) a2a, $\lambda = 10^{-3}$ (h) a2a, $\lambda = 10^{-4}$

Figure 3: Comparison of NL1, NL2 with Newton's method in terms of communication complexity.



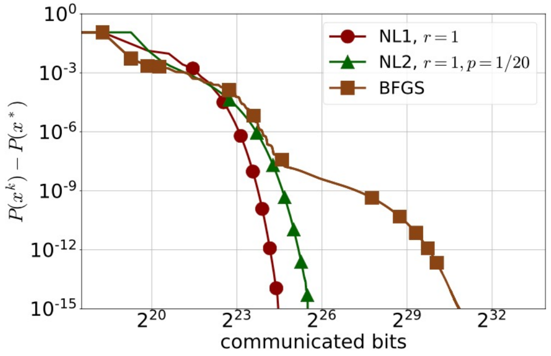
(a) w8a, $\lambda = 10^{-3}$

NL1 & NL2

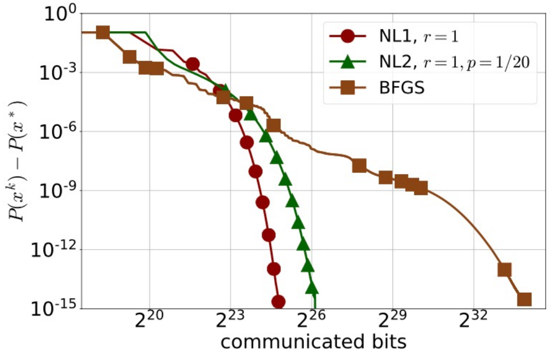
vs

BFGS

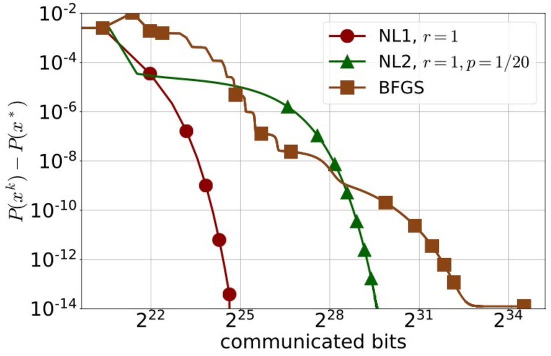
NL1 & NL2 vs BFGS



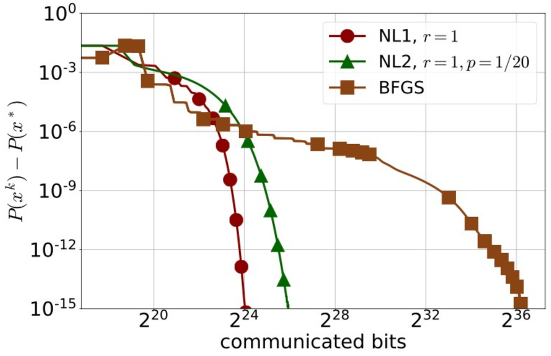
(a) a9a, $\lambda = 10^{-3}$



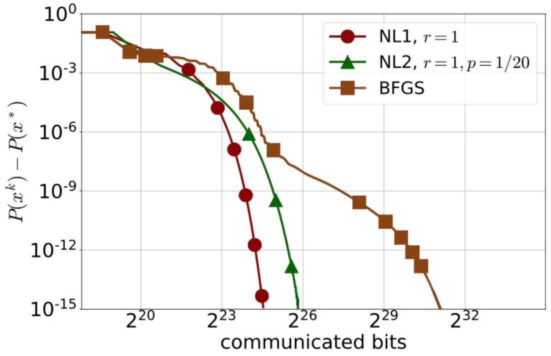
(b) a9a, $\lambda = 10^{-4}$



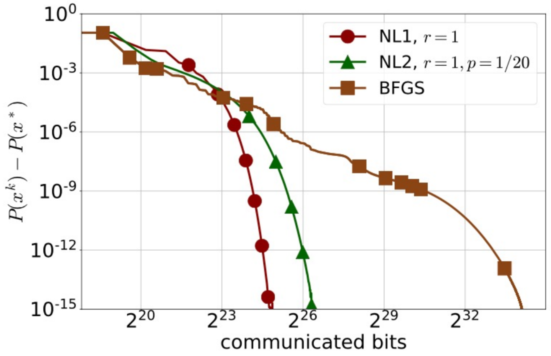
(c) w8a, $\lambda = 10^{-3}$



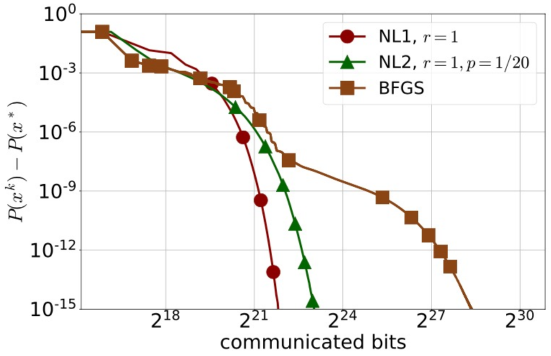
(d) phishing, $\lambda = 10^{-5}$



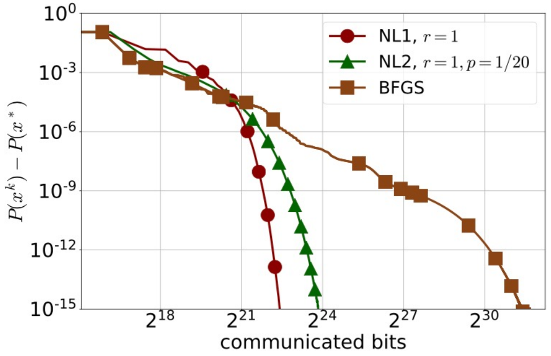
(e) a7a, $\lambda = 10^{-3}$



(f) a7a, $\lambda = 10^{-4}$

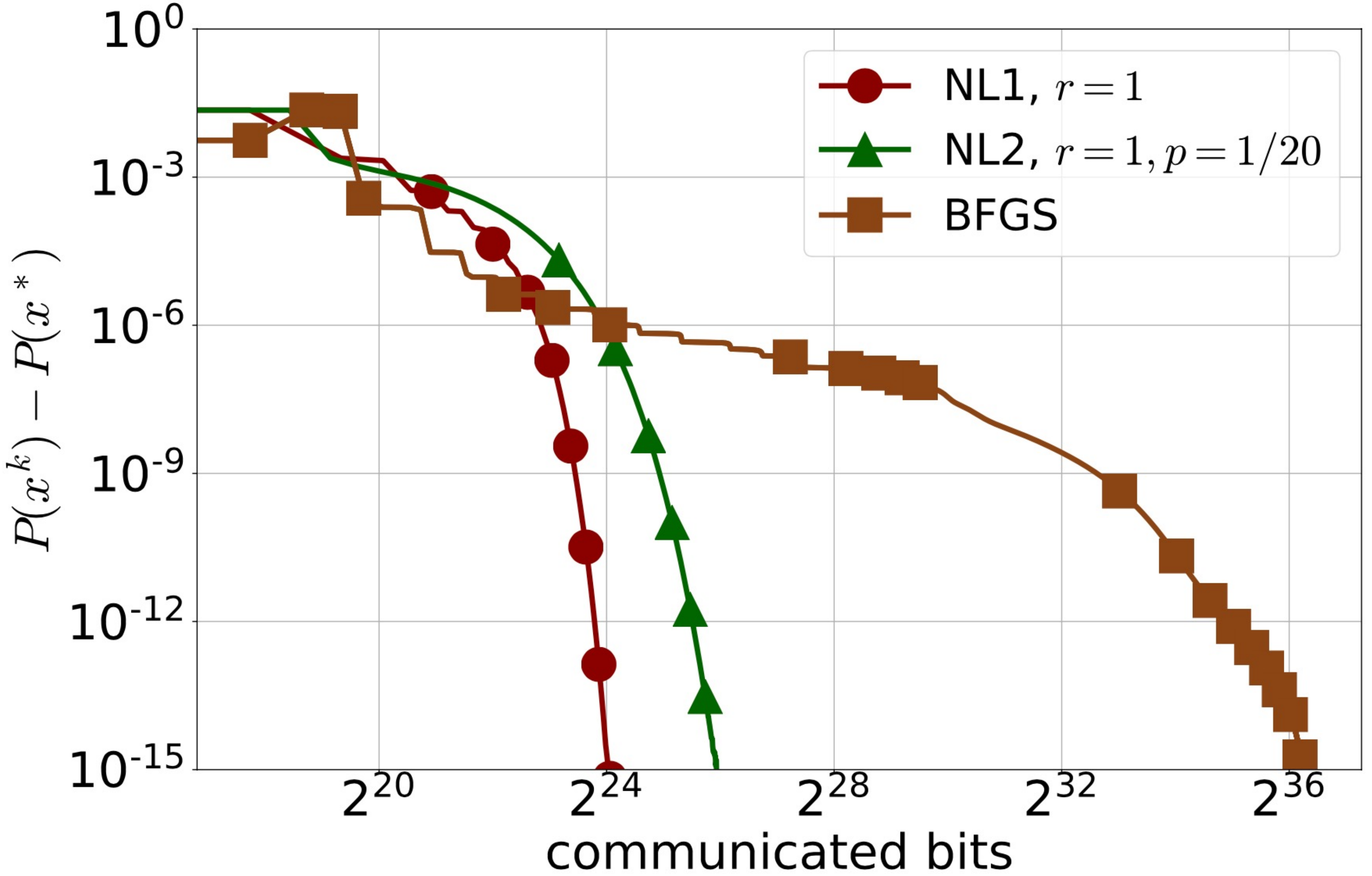


(g) a2a, $\lambda = 10^{-3}$



(h) a2a, $\lambda = 10^{-4}$

Figure 4: Comparison of NL1, NL2 and BFGS in terms of communication complexity.



(d) $\texttt{phishing}$, $\lambda = 10^{-5}$

NL1 & NL2

vs

Accelerated DIANA

Zhize Li, Dmitry Kovalev, Xun Qian and Peter Richtárik
Acceleration for compressed gradient descent in distributed and federated optimization
ICML, 2020

NL1 & NL2 vs ADIANA

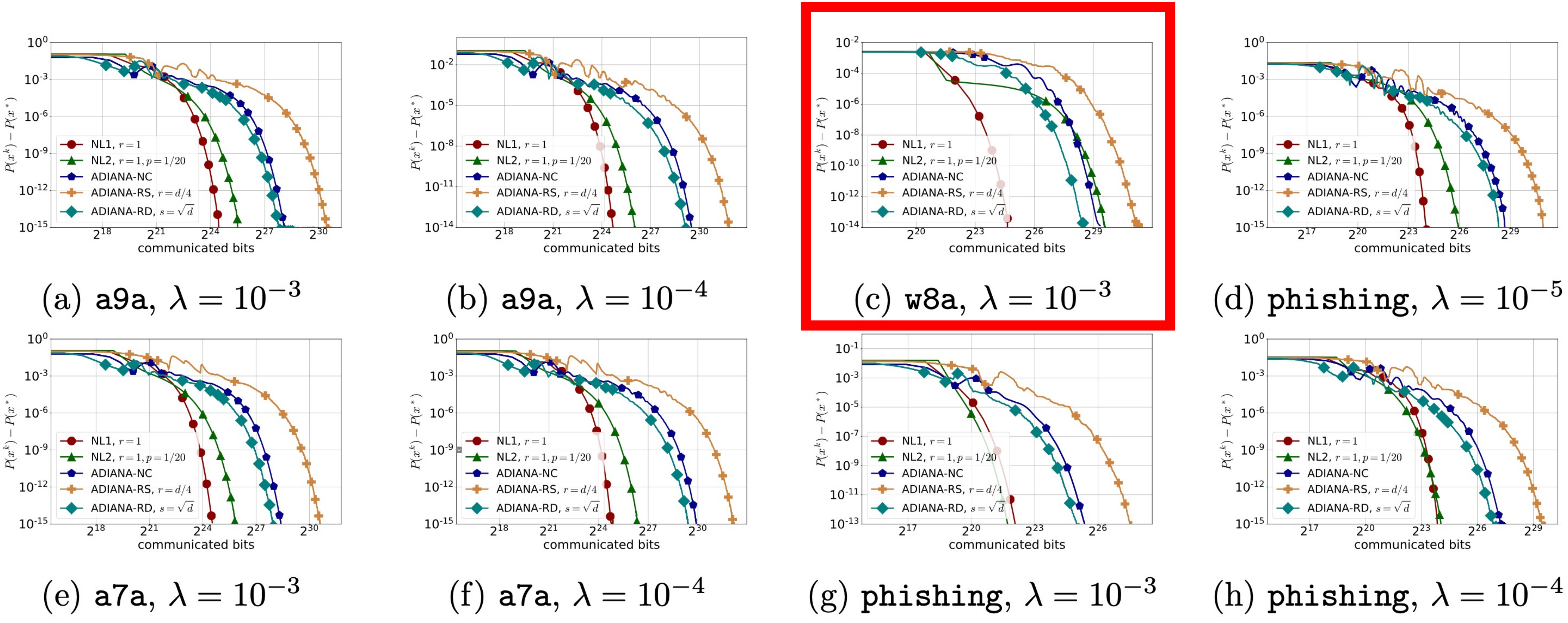
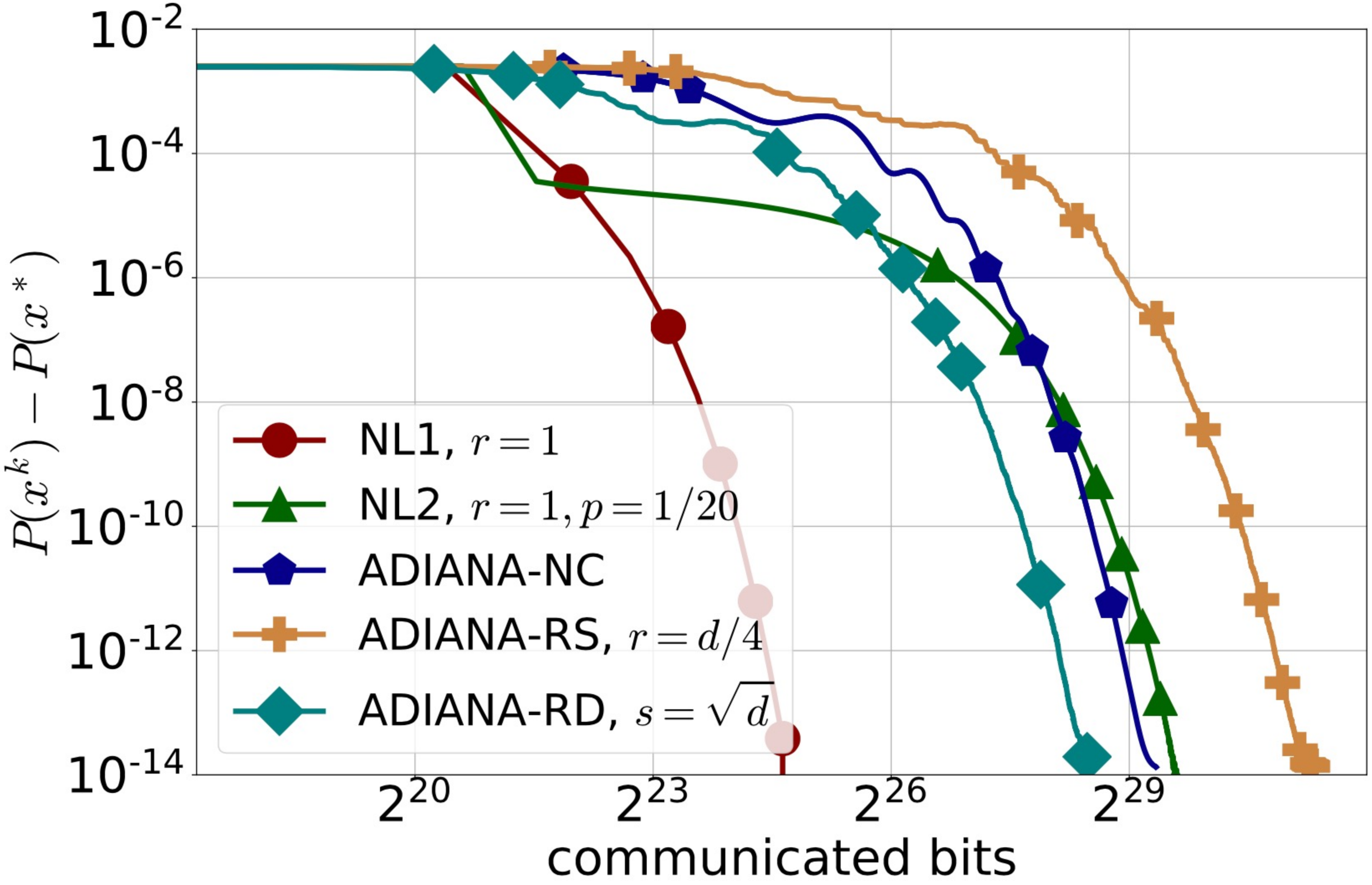


Figure 5: Comparison of NL1, NL2 with ADIANA in terms of communication complexity.



(c) w8a, $\lambda = 10^{-3}$

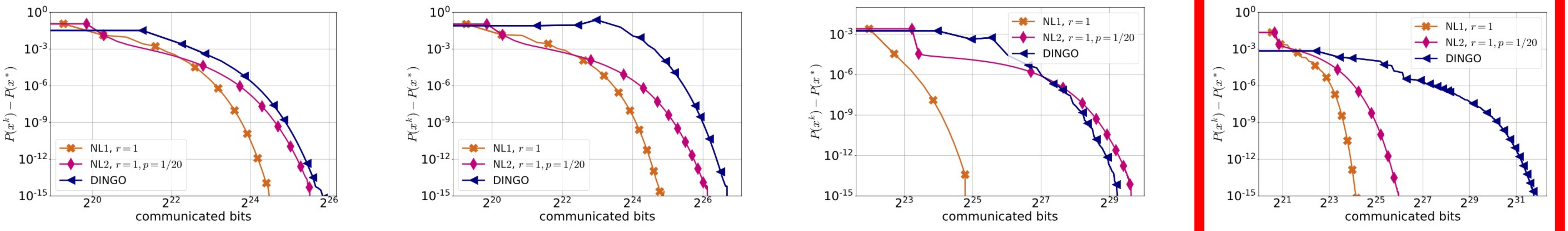
NL1 & NL2

vs

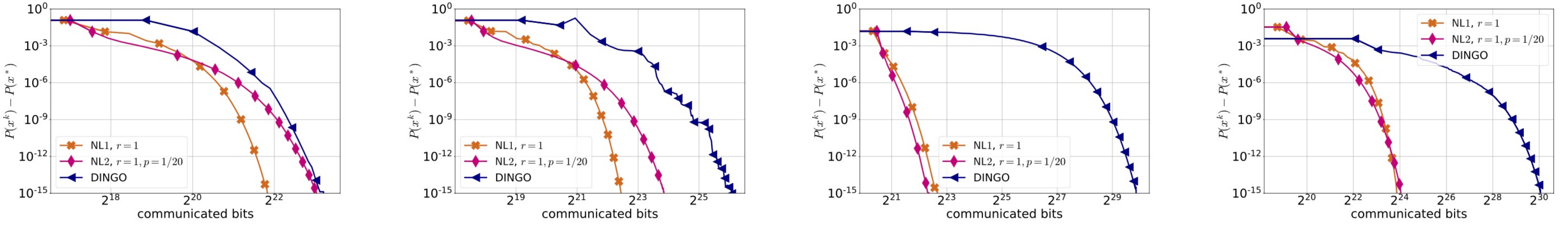
DINGO

Rixon Crane and Fred Roosta
DINGO: Distributed Newton-type method for gradient-norm optimization
NeurIPS, 2019

NL1 & NL2 vs DINGO

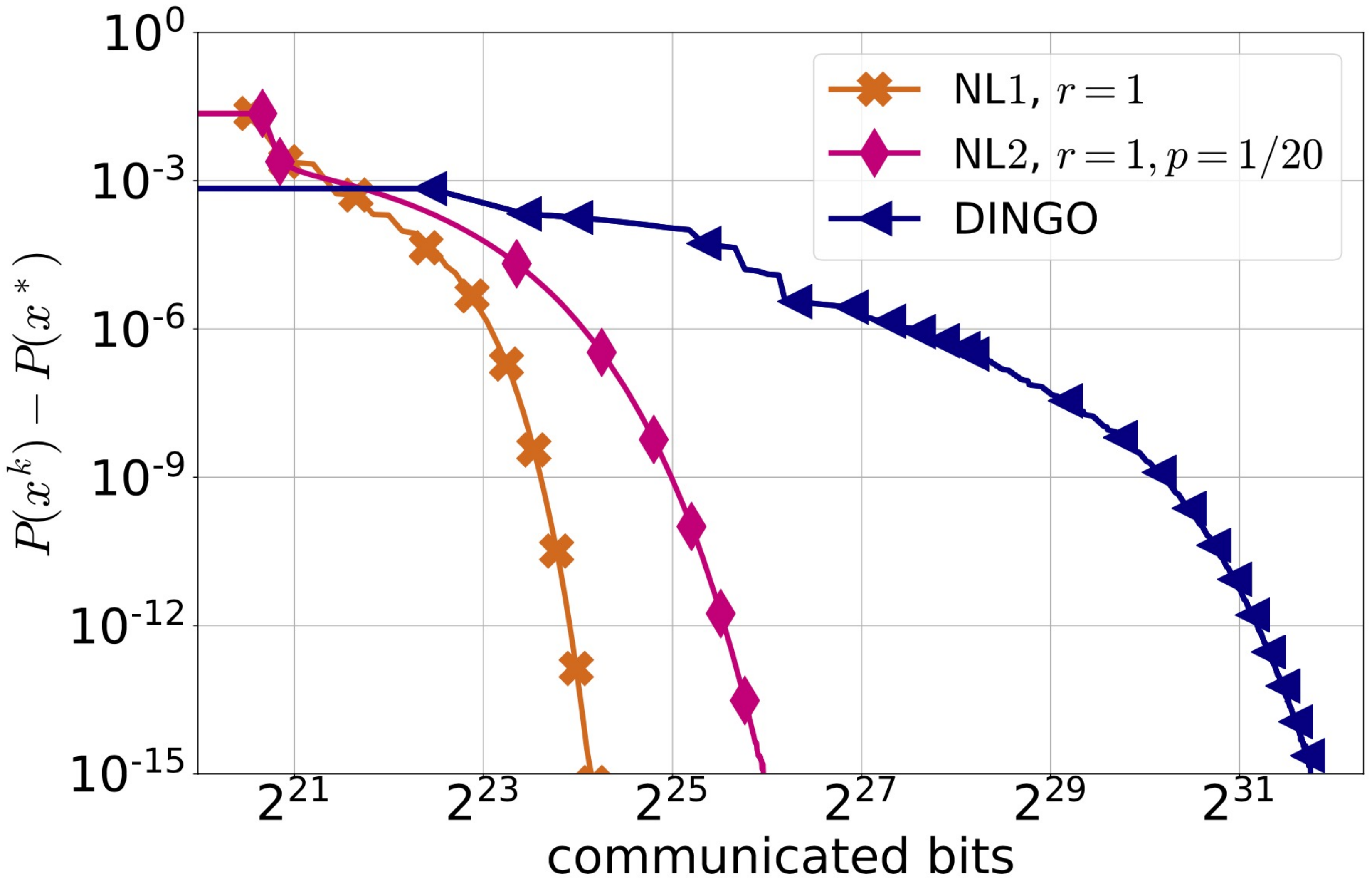


(a) **a9a**, $\lambda = 10^{-3}$ (b) **a9a**, $\lambda = 10^{-4}$ (c) **w8a**, $\lambda = 10^{-3}$ (d) **phishing**, $\lambda = 10^{-5}$



(e) **a2a**, $\lambda = 10^{-3}$ (f) **a2a**, $\lambda = 10^{-4}$ (g) **phishing**, $\lambda = 10^{-3}$ (h) **phishing**, $\lambda = 10^{-4}$

Figure 6: Comparison of NL1, NL2 with DINGO in terms of communication complexity.



(d) phishing, $\lambda = 10^{-5}$

The End

8. On DIANA & Friends

Our Hessian Learning Mechanism is Inspired by DIANA

Filip Hanzely, Konstantin Mishchenko and Peter Richtárik
SEGA: Variance reduction via gradient sketching
NeurIPS, 2018

SEGA ≈ **"Single node"** DIANA

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč and Peter Richtárik
Distributed learning with compressed gradient differences
arXiv:1901.09269, 2019

Original DIANA paper

Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Peter Richtárik and Sebastian Stich
Stochastic distributed learning with gradient quantization and variance reduction
arXiv:1904.05115, 2019

Generalized DIANA:
- Any unbiased compressor
- Variance reduction for finite-sum on machines (VR-DIANA)

Eduard Gorbunov, Filip Hanzely and Peter Richtárik
A unified theory of SGD: variance reduction, sampling, quantization and coordinate descent
AISTATS, 2020

General analysis of many SGD methods in a single theorem, including DIANA

Sélim Chraibi, Ahmed Khaled, Dmitry Kovalev, Adil Salim, Peter Richtárik and Martin Takáč
Distributed fixed point methods with compressed iterates
arXiv:1912.09925, 2019

DIANA for **fixed point problems**

Our Hessian Learning Mechanism is Inspired by DIANA

Zhize Li, Dmitry Kovalev, Xun Qian and Peter Richtárik
Acceleration for compressed gradient descent in distributed and federated optimization
ICML, 2020

Accelerated DIANA
(ADIANA)

Zhize Li and Peter Richtárik
A unified analysis of stochastic gradient methods for nonconvex federated optimization
SpicyFL 2020: NeurIPS Workshop on Scalability, Privacy, and Security in Federated Learning

Unified analysis of distributed compressed gradient methods for **nonconvex** functions, including DIANA

Eduard Gorbunov, Dmitry Kovalev, Dmitry Makarenko, and Peter Richtárik
Linearly converging error compensated SGD
NeurIPS, 2020

DIANA for **Error Compensation**
(EC-SGD-DIANA, EC-LSVRG-DIANA)

Dmitry Kovalev, Anastasia Koloskova, Martin Jaggi, Peter Richtárik, and Sebastian U. Stich
A linearly convergent algorithm for decentralized optimization: sending less bits for free!
AISTATS, 2021

Decentralized DIANA

Mher Safaryan, Filip Hanzely and Peter Richtárik
Smoothness matrices beat smoothness constants: better communication compression techniques for distributed optimization
arXiv:2102.07245, 2021

DIANA and ADIANA benefit from **matrix smoothness**
(DIANA+, ADIANA+)

Our Hessian Learning Mechanism is Inspired by DIANA

Eduard Gorbunov, Konstantin Burlachenko, Zhize Li and Peter Richtárik
MARINA: faster non-convex distributed learning with compression
arXiv:2102.07845, 2021

MARINA

- Inspired by DIANA, but compressing **true gradient differences**
- Uses a **biased estimator**
- Current **theoretical SOTA** among communication efficient distributed methods for nonconvex problems (better than DIANA, which was previous SOTA)

9. FedNL

Mher Safaryan, Rustem Islamov, Xun Qian and Peter Richtárik
FedNL: Making Newton-type Methods Applicable to Federated Learning
arXiv:2106.02969, 2021

Improvements Over the First Paper

Table 1: Comparison of the main features of our family of FedNL algorithms and results with those of Islamov et al. [2021], which we used as an inspiration. We have made numerous and significant modifications and improvements in order to obtain methods applicable to federated learning.

#	Feature	Islamov et al. [2/'21]	This Work [5/'21]
[hd]	supports heterogeneous data setting	✓	✓
[fs]	applies to general finite-sum problems	✗	✓
[as]	uses adaptive stepsizes	✓	✓
[pe]	privacy is enhanced (training data is not sent to the server)	✗	✓
[uc]	supports unbiased Hessian compression (e.g., Rand-K)	✓	✓
[cc]	supports contractive Hessian compression (e.g., Top-K)	✗	✓
[fr]	fast local rate: independent of the condition number	✓	✓
[fr]	fast local rate: independent of the # of training data points	✗	✓
[fr]	fast local rate: independent of the compressor variance	✗	✓
[pp]	supports partial participation	✗	✓(Alg 2)
[gg]	has global convergence guarantees via line search	✗	✓(Alg 3)
[gg]	has global convergence guarantees via cubic regularization	✓	✓(Alg 4)
[gc]	supports smart uplink gradient compression at the devices	✗	✓(Alg 5)
[mc]	supports smart downlink model compression by the master	✗	✓(Alg 5)
[lc]	performs useful local computation	✓	✓

1st paper 2nd paper

Summary of Complexity Results

Table 2: Summary of algorithms proposed and convergence results proved in this paper.

Method	Convergence result †	Convergence type	Convergence rate	Rate independent of the condition # (left) # training data (middle) compressor (right)	Theorem
Newton Zero N0 (Equation (9))	$r_k \leq \frac{1}{2^k} r_0$	local	linear	✓ ✓ ✓	3.6
FedNL (Algorithm 1)	$r_k \leq \frac{1}{2^k} r_0$	local	linear	✓ ✓ ✓	3.6
	$\Phi_1^k \leq \theta^k \Phi_1^0$	local	linear	✓ ✓ ✗	3.6
	$r_{k+1} \leq c\theta^k r_k$	local	superlinear	✓ ✓ ✗	3.6
Partial Participation FedNL-PP (Algorithm 2)	$\mathcal{W}^k \leq \theta^k \mathcal{W}^0$	local	linear	✓ ✓ ✓	C.1
	$\Phi_2^k \leq \theta^k \Phi_2^0$	local	linear	✓ ✓ ✗	C.1
	$r_{k+1} \leq c\theta^k \mathcal{W}_k$	local	linear	✓ ✓ ✗	C.1
Line Search FedNL-LS (Algorithm 3)	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	D.1
Cubic Regularization FedNL-CR (Algorithm 4)	$\Delta_k \leq c/k$	global	sublinear	✗ ✓ ✓	E.1
	$\Delta_k \leq \theta^k \Delta_0$	global	linear	✗ ✓ ✓	E.1
	$\Phi_1^k \leq \theta^k \Phi_1^0$	local	linear	✓ ✓ ✗	E.1
	$r_{k+1} \leq c\theta^k r_k$	local	superlinear	✓ ✓ ✗	E.1
Bidirectional Compression FedNL-BC (Algorithm 5)	$\Phi_3^k \leq \theta^k \Phi_3^0$	local	linear	✓ ✓ ✗	F.4
Newton Star NS (Equation (55))	$r_{k+1} \leq c r_k^2$	local	quadratic	✓ ✓ ✓	G.1

Quantities for which we prove convergence: (i) distance to solution $r_k := \|x^k - x^*\|^2$; $\mathcal{W}^k := \frac{1}{n}\sum_{i=1}^n \|w_i^k - x^*\|^2$ (ii) Lyapunov functions $\Phi_1^k := c\|x^k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2$; $\Phi_2^k := c\mathcal{W}^k + \frac{1}{n}\sum_{i=1}^n \|\mathbf{H}_i^k - \nabla^2 f_i(x^*)\|_{\rm F}^2$; $\Phi_3^k := \|z^k - x^*\|^2 + c\|w^k - x^*\|^2$. (iii) Function value suboptimality $\Delta_k := f(x^k) - f(x^*)$

† constants $c > 0$ and $\theta \in (0,1)$ are possibly different each time they appear in this table. Refer to the precise statements of the theorems for the exact values.

The End
(For Real)