Newton Method over Networks is Fast up to the Statistical Precision

Amir Daneshmand [1] **Gesualdo Scutari** [1] **Pavel Dvurechensky** [2 3] **Alexander Gasnikov** [4 3]

Abstract

We propose a distributed cubic regularization of the Newton method for solving (constrained) empirical risk minimization problems over a network of agents, modeled as undirected graph. The algorithm employs an *inexact, preconditioned* Newton step at each agent's side: the gradient of the centralized loss is iteratively estimated via a gradient-tracking consensus mechanism and the Hessian is subsampled over the local data sets. No Hessian matrices are thus exchanged over the network. We derive global complexity bounds for convex and strongly convex losses. Our analysis reveals an interesting interplay between sample and iteration/communication complexity: *statistically accurate* solutions are achievable roughly in the same number of iterations of the centralized cubic Newton, with a communication cost per iteration of the order of $\widetilde{\mathcal{O}}\big(1/\sqrt{1-\rho}\big)$, where ρ characterizes the connectivity of the network. This represents a significant communication saving with respect to that of existing, statistically oblivious, distributed Newton-based methods over networks.

1. Introduction

We study Empirical Risk Minimization (ERM) problems over a network of m agents, modeled as undirected graph. Differently from master/slave systems, no centralized node is assumed in the network (which will be referred to as *meshed* network). Each agent i has access to n i.i.d. samples $z_i^{(1)}, \ldots, z_i^{(n)}$ drawn from an unknown, common distribution on $\mathcal{Z} \subseteq \mathbb{R}^p$; the associated empirical risk reads

$$f_i(x) \triangleq \frac{1}{n}\sum_{j=1}^{n} \ell\big(x; z_i^{(j)}\big), \tag{1}$$

where $\ell : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is the loss function, assumed to be (strongly) convex in x, for any given $z \in \mathcal{Z}$. Agents aim to minimize the total empirical risk over the $N = mn$ samples, resulting in the following ERM over networks:

$$\widehat{x} \in \operatorname*{argmin}_{x\in\mathcal{K}} F\left(x\right) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_i\left(x\right), \tag{P}$$

where $\mathcal{K} \subseteq \mathbb{R}^d$ is convex and known to the agents.

Since the functions f_i can be accessed only locally and routing local data to other agents is infeasible or highly inefficient, solving (P) calls for the design of distributed algorithms that alternate between a local computation procedure at each agent's side, and a round of communication among neighboring nodes. The cost of communications is often considered the bottleneck for distributed computing, if compared with local (possibly parallel) computations (e.g., (Bekkerman et al., 2011; Lian et al., 2017)). Therefore, our goal is developing *communication-efficient* distributed algorithms that solve (P) within the *statistical* precision.

The provably faster convergence rates of second order methods over gradient-based algorithms make them potential candidates for communication saving (at the cost of more computations). Despite the success of Newton-like methods to solve ERM in a centralized setting (e.g., (Mokhtari et al., 2016a; Bottou et al., 2018)), including master/slave architectures (Zhang & Xiao, 2015; Shamir et al., 2014; Ma & Takac, 2017; Jahani et al., 2020; Soori et al., 2020), their distributed counterparts on meshed networks are not on par: convergence rates provably faster than those of first order methods are achieved at high communication costs (Uribe & Jadbabaie, 2020a; Zhang et al., 2020), cf. Sec. 1.2.

We claim that stronger guarantees of second order methods over meshed networks can be certified if a *statistically-informed* design/analysis is put forth, in contrast with statistically agnostic approaches that look at (P) as deterministic optimization and target any arbitrarily small suboptimality. To do so, we build on the following two key insights.

- **Fact 1 (statistical accuracy):** When it comes to learning problems, the ERM (P) is a surrogate of the population minimization

$$x^\star \in \operatorname*{argmin}_{x\in\mathcal{K}} F_P\left(x\right) \triangleq \mathbb{E}_{Z\sim\mathbb{P}}\,\ell\left(x; Z\right). \tag{2}$$

The ultimate goal is to estimate $x^\star$ via the ERM (P). Denoting by $x_\varepsilon \in \mathcal{K}$ the estimate returned by the algorithm, we

[1] School of Industrial Engineering, Purdue University, West-Lafayette, IN, USA [2] Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany [3] Higher School of Economics (HSE) University, Moscow, Russia [4] Moscow Institute of Physics and Technology, Dolgoprudny, Russia. Correspondence to: Gesualdo Scutari <gscutari@purdue.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

have the risk decomposition (neglecting the approximation error due to the use of a specific set of models $x \in \mathcal{K}$):

$$\begin{aligned} & F_P(x_\varepsilon) - F_P(x^\star) \\ & = \underbrace{\left\{F_P(x_\varepsilon) - F(x_\varepsilon)\right\}}_{\leq \text{statistical error}} + \left\{F(x_\varepsilon) - F(x^\star)\right\} \\ & \quad + \underbrace{\left\{F(x^\star) - F_P(x^\star)\right\}}_{\leq \text{statistical error}} \\ & \leq \mathcal{O}(\text{statistical error}) + \underbrace{\left\{F(x_\varepsilon) - F(\widehat{x})\right\}}_{= \text{optimization error}} \end{aligned} \tag{3}$$

where the statistical error is usually of the order $\mathcal{O}(1/\sqrt{N})$ or $\mathcal{O}(1/N)$ (cf. Sec. 2). It is thus sufficient to reach an optimization accuracy $F(x_\varepsilon) - F(\widehat{x}) = \mathcal{O}(\text{statistical error})$. This can potentially save communications.

• **Fact 2 (statistical similarity):** Under mild assumptions on the loss functions and i.i.d samples across the agents (e.g., (Zhang & Xiao, 2015; Hendrikx et al., 2020b)), it holds with high probability (and uniformly on $\mathcal{Z}$)

$$\left\|\nabla^2 f_i(x) - \nabla^2 F(x)\right\| \leq \beta = \widetilde{\mathcal{O}}(1/\sqrt{n}), \quad \forall x \in \mathcal{K}, \tag{4}$$

with $\widetilde{\mathcal{O}}$ hiding log-factors and the dependence on d. In words, the local empirical losses f_i are statistically similar to each other and the average F, especially for large n.

The key insight of Fact 1 is that one can target suboptimal solutions of (P) within the statistical error. This is different from seeking a distributed optimization method that achieves any arbitrarily small empirical suboptimality. Fact 2 suggests a further reduction in the communication complexity via *statistical preconditioning*, that is, subsampling the Hessian of F over the local data sets, so that no Hessian matrix has to be transmitted over the network. This paper shows that, if synergically combined, these two facts can improve the communication complexity of distributed second order methods over meshed networks.

1.1. Major contributions

We propose and analyze a decentralization of the cubic regularization of the Newton method (Nesterov & Polyak, 2006) over meshed networks. The algorithm employs a local computation procedure performed in parallel by the agents coupled with a round of (perturbed) consensus mechanisms that aim to track *locally* the gradient of F (a.k.a. gradient-tracking) as well as enforce an agreement on the local optimization directions. The optimization procedure is an inexact, preconditioned (cubic regularized) Newton step whereby the gradient of F is estimated by gradient tracking while the Hessian of F is subsampled over the local data sets. Neither a line-search nor communication of Hessian matrices over the network are performed.

We established for the first time *global* convergence for different classes of ERM problems, as summarized in Table 1. Our results are of two types: i) classical complexity analysis (number of communication rounds) for arbitrary solution accuracy (right panel); ii) and complexity bounds for statistically optimal solutions (left panel, V_N is the statistical error). Postponing to Sec 4 a detailed discussion of these results, here we highlight some key novelties of our findings. **Convex ERM:** For convex F, if arbitrary ε-solutions are targeted, the algorithm exhibits a two-speed behavior: 1) a first rate of the order of $\mathcal{O}((1/\sqrt{1-\rho}) \cdot \sqrt{LD^3/\varepsilon^{1+\alpha}})$, as long as $\varepsilon = \Omega(LD^3\beta^2)$; up to the network dependent factor $1/\sqrt{1-\rho}$, this (almost) matches the rate of the centralized Newton method (Nesterov & Polyak, 2006); and 2) the slower rate $\mathcal{O}((1/\sqrt{1-\rho}) \cdot (LD^3\beta^2)/\varepsilon)$, which is due to the local subsampling of the global Hessian $\nabla^2 F$; this term is dominant for smaller values of ε. The interesting fact is that $\varepsilon = \Omega(LD^3\beta^2)$ is of the order of the statistical error V_N. Therefore, rates of the order of centralized ones are provably achieved up to statistical precision (left panel). **Strongly Convex ERM ($\beta < \mu$):** The communication complexity shows a three-phase rate behaviour (right panel); for arbitrarily small $\varepsilon > 0$, the worst-case communication complexity is linear, of the order of $\widetilde{\mathcal{O}}\big((1/\sqrt{1-\rho}) \cdot (\beta/\mu) \cdot \log(1/\varepsilon)\big)$. Faster rates are certified when $\varepsilon = \mathcal{O}(V_N)$ (left panel). Note that the region of superlinear convergence is a false improvement when the first term $m^{1/4}\sqrt{LD/\mu}$ is dominant, e.g., F is ill-conditioned and n is not large. This term is unavoidable (Nesterov & Polyak, 2006)–unless more refined function classes are considered, such as self-concordant or quadratic ($L = 0$). The left panel shows improved rates in the latter case or under an initialization within a $\mathcal{O}(1/\sqrt{n})$-neighborhood of the solution. **Strongly Convex ERM ($\beta \geq \mu$):** This is a common setting when F_P is convex and a regularizer is used in the ERM (P) for learnability/stability purposes; typically, $\mu = \mathcal{O}(1/\sqrt{N})$, $\beta = \mathcal{O}(1/\sqrt{n})$. We proved linear rate for arbitrary ε-values. Differently from the majority of first-order methods over meshed networks (cf. Sec. 1.2), this rate does not depend on the condition number of F but on the generally more favorable ratio β/μ. Furthermore, when $\varepsilon = \mathcal{O}(V_N)$, the rate does not improve over the convex case. In summary, we propose a second-order method solving convex and strongly convex problems over meshed networks that, for the first time, enjoys global complexity bounds and communication complexity close to oracle complexity of centralized methods up to the statistical precision.

1.2. Related Works

The literature of distributed optimization is vast; here we review relevant methods applicable to *meshed networks*, omitting the less relevant work considering only master-slave systems (a.k.a star networks).

• **Statistically oblivious methods:** Despite being vast and providing different communication and oracle complexity bounds, the literature (e.g., (Jakovetić et al., 2014; Shi et al., 2015; Arjevani & Shamir, 2015; Nedic et al., 2017; Sca-

Table 1. Communication complexity of DiRegINA to $\varepsilon > 0$ suboptimality for (strongly) convex ERM. **Right column:** arbitrary ε values. **Left column:** $\varepsilon = \Omega(V_N)$, V_N is the statistical error [cf. (3)]. The other parameters are: μ and L are the strong convexity constant of F and Lipschitz constant of $\nabla^2 F$, respectively; D and D_p are estimates of the optimality gap at the initial point; β measures the similarity of $\nabla^2 f_i$ [cf. (4)]; ρ characterizes the connectivity of the network; and $\alpha > 0$ is an arbitrarily small constant.

Problem		$\varepsilon = \Omega(V_N)$ **(statistical error)**	$\varepsilon > 0$ **(arbitrary)**
Convex $\mu = 0$ $V_N = \mathcal{O}(1/\sqrt{N})$	Thm. 7 Cor. 8	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \sqrt{\frac{LD^3}{V_N^{1+\alpha}}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \left\{\sqrt{\frac{LD^3}{\varepsilon^{1+\alpha}}} + \frac{LD^3\beta}{\varepsilon^{1+\alpha/2}}\right\}\right)$
Strongly-convex $0 < \beta < \mu$ $V_N = \mathcal{O}(1/N)$ $\mu = \mathcal{O}(1)$	$L > 0$ Thm. 9 Cor.10	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{m^{1/4}\sqrt{\frac{LD}{\mu}} + \log\log\left(\frac{\mu^3}{mL^2V_N}\right)\right\}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{m^{1/4} \cdot \sqrt{\frac{LD}{\mu}} + \log\log\left(\frac{\mu^2}{\beta^2}\min\left\{1, \frac{\beta^2\mu}{mL^2} \cdot \frac{1}{\varepsilon}\right\}\right) + \frac{\beta}{\mu}\log\left(\frac{\beta^2\mu}{mL^2\varepsilon}\right)\right\}\right)$
	$L > 0$ +initialization (Cor. 11)	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \left\{\log\log\left(\frac{\mu^3}{mL^2} \cdot \frac{1}{V_N}\right)\right\}\right), \quad \beta = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{\log\log\left(\frac{\mu^2}{\beta^2} \cdot \min\left(1, \frac{\beta^2\mu}{mL^2\varepsilon}\right)\right) + \frac{\beta}{\mu}\log\left(\frac{\beta^2\mu}{mL^2\varepsilon}\right)\right\}\right)$
	$L = 0$ (Thm. 18, appendix E.4)	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \log\log\left(\frac{D_p}{V_N}\right)\right), \quad \beta = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \left\{\log\log\left(\frac{D_p}{\varepsilon}\right) + \frac{\beta}{\mu}\log\left(\frac{D_p\beta^2}{\mu^2\varepsilon}\right)\right\}\right)$
Strongly-convex (regularized) $0 < \mu \leq \beta$ $V_N = \mathcal{O}(1/\sqrt{N})$	$L > 0$ (Thm. 12)	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}m^{1/2}\sqrt{\frac{LD}{V_N}}\right), \quad \begin{cases}\mu = \mathcal{O}(V_N)\\ \beta = \mathcal{O}(\frac{1}{\sqrt{n}})\end{cases}$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{\sqrt{\frac{LD}{\mu}}\left(1 + m^{1/4}\sqrt{\frac{\beta}{\mu}}\right) + \frac{\beta}{\mu}\log\left(\frac{\beta^2\mu}{mL^2\varepsilon}\right)\right\}\right)$
	$L = 0$ (Thm. 19, appendix G)	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot m^{1/2} \cdot \log\left(\frac{1}{V_N}\right)\right), \quad \begin{cases}\mu = \mathcal{O}(V_N)\\ \beta = \mathcal{O}(\frac{1}{\sqrt{n}})\end{cases}$	$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \frac{\beta}{\mu} \cdot \log\left(\frac{1}{\varepsilon}\right)\right)$

man et al., 2017; Lan et al., 2017; Uribe et al., 2020; Rogozin et al., 2020)) on decentralized **first-order methods** for minimizing Q-Lipschitz-smooth and μ-strongly convex global objective F mostly focuses on the particular case where $n = 1$ in (1) and $\mathcal{K} = \mathbb{R}^d$, and does not take into account statistical similarity of the risks. The best convergence results for nonaccelerated first-order methods certify linear rate, scaling with the condition number $\kappa = Q/\mu$ (Q is the Lipschitz constant of ∇F); Nesterov-based acceleration improves the dependence to $\sqrt{\kappa}$ (Gorbunov et al., 2020). This performance can still be unsatisfactory when $1 + \beta/\mu < \kappa$ (resp. $1 + \beta/\mu < \sqrt{\kappa}$). This is the typical situation of ill-conditioned problems, such as many learning problems where the regularization parameter that is optimal for test predictive performance is very small (Hendrikx et al., 2020b). For instance, consider the ridge-regression problem with optimal regularization parameter $\mu = 1/\sqrt{mn}$ (Table 1 in (Zhang & Xiao, 2015)), we have: $\kappa = \mathcal{O}(\sqrt{m \cdot n})$ while $\beta/\mu = \mathcal{O}(\sqrt{m})$. Notice that the former grows with the local sample size n, while the latter is independent.

A number of **second-order methods** were proposed for distributed optimization over meshed networks, with typical results being local superlinear convergence (Jadbabaie et al., 2009; Wei et al., 2013; Tutunov et al., 2019) or global linear convergence no better than that of first-order methods (Mokhtari et al., 2016d; 2017; 2016c; Eisen et al., 2019; Jiaojiao et al., 2020). Improved upon first-order methods global bounds are achieved by exploiting expensive sending local Hessians over the network–such as (Zhang et al., 2020), obtaining communication complexity bound $\mathcal{O}((mL\|\nabla f(x_0)\|/\mu^2) + \log\log(1/\varepsilon))$–or employing double-loop schemes (Uribe & Jadbabaie, 2020b) wherein at each iteration, a distributed first-order method is called to find the Newton direction, obtaining iteration complexity $\mathcal{O}(\sqrt[3]{LD^3/\varepsilon})$ at the price of excessive communications per iteration. Furthermore, these schemes cannot handle constraints. To the best of our knowledge, no distributed second-order method over meshed networks has been proved to globally converge with communication complexity bounds even up to a network dependent factor close to the standard (Nesterov & Polyak, 2006) bounds $\mathcal{O}(\sqrt{(LD^3)/\varepsilon})$ for convex and $\mathcal{O}(\sqrt{LD/\mu} + \log\log(\mu^3/L^2\varepsilon))$ for μ-strongly convex problems. Table 1 shows the first results of this genre.

• **Methods exploiting statistical similarity:** Starting the works (Shamir et al., 2014; Arjevani & Shamir, 2015) several papers studied the idea of statistical preconditioning to decrease the communication complexity over star networks, for different problem classes; example include (Shamir et al., 2014; Reddi et al., 2016; Yuan & Li, 2019) (quadratic losses), (Zhang & Xiao, 2015) (self-concordant losses), (Wang et al., 2018) (under $n > d$), and (Fan et al., 2019) (composite optimization), with (Hendrikx et al., 2020b; Dvurechensky et al., 2021) employing acceleration. None of these methods are implementable over meshed networks, because they rely on a centralized (master) node. To our knowledge, Network-DANE (Li et al., 2019) and SONATA (Sun et al., 2019) are the only two methods that leverage statistical similarity to enhance convergence of distributed methods over meshed networks; (Li et al., 2019) studies strongly convex quadratic losses while (Sun et al., 2019) considers general objectives, achieving a communication complexity of $\widetilde{\mathcal{O}}((1/\sqrt{1-\rho}) \cdot \beta/\mu \cdot \log(1/\varepsilon))$. Both schemes call at every iteration for an exact solution of local strongly convex problems while our subproblems are based on second-order approximations, computationally thus less demanding. Nevertheless, our algorithm retains same rate dependence on β/μ. Our study covers also non-strongly convex losses.

2. Setup and Background

2.1. Problem setting

We study convex and strongly convex instances of the ERM (P); specifically, we make the following assumptions [note that, although explicitly omitted, each $f_i(x)$ and thus F depend on the sample $z \in \mathcal{Z}$ via $\ell(x, z)$; all the assumptions below are meant to hold uniformly on $\mathcal{Z}$].

Assumption 1 (convex ERM). *The following hold:*

(i) *$\emptyset \neq \mathcal{K} \subseteq \mathbb{R}^d$ is closed and convex;*

(ii) *Each $f_i : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ is twice differentiable and μ_i-strongly convex on (an open set containing) $\mathcal{K}$, with $\mu_i \geq 0$;*

(iii) *Each ∇f_i is Q_i-Lipschitz continuous on $\mathcal{K}$, where ∇f_i is the gradient with respect to x; let $Q_{\max} \triangleq \max_{i=1,\ldots,m} Q_i$;*

(iv) *F has bounded level sets.*

Assumption 2 (strongly convex ERM). *Assumption 1(i)-(iii) holds and F is μ-strongly convex on $\mathcal{K}$, with $\mu > 0$.*

The following condition is standard when studying second order methods.

Assumption 3. *$\nabla^2 F : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is L-Lipschitz continuous on $\mathcal{K}$, i.e., $\left\|\nabla^2 F(x) - \nabla^2 F(y)\right\| \leq L\|x - y\|$, for some $L > 0$ and all $x, y \in \mathcal{K}$.*

Statistical accuracy: As anticipated in Sec. 1, we are interested in computing estimates of the population minimizer (2) up to the statistical error using the ERM rule (3). To do so, throughout the paper, we postulate the following standard uniform convergence property, which suffices for learnability by (3): there exists a constant V_N, dependent on $N = m\,n$, such that

$$\sup_{x\in\mathcal{K}} |F(x) - F_P(x)| \leq V_N \quad \text{w.h.p.} \tag{5}$$

The statistical accuracy V_N has been widely studied in the literature, e.g., (Vapnik, 2013; Bousquet, 2002; Bartlett et al., 2006; Frostig et al., 2015; Shai & Ben-David, 2014). Consistently with these works, we will assume:

1. $V_N = \mathcal{O}(1/N)$, for μ-strongly convex F and F_P, with $0 < \mu = \mathcal{O}(1)$;
2. $V_N = \mathcal{O}(1/\sqrt{N})$ for convex or μ-strongly convex F, with $\mu = \mathcal{O}(1/\sqrt{N})$.

These cases cover a variety of problems of practical interest. An example of case 1 is a loss in the form $\ell(x; z) = f(x; z) + (\mu/2)\|x\|^2$, with fixed regularization parameter and f convex in x (uniformly on z), as in ERM of linear predictors for supervised learning (Sridharan et al., 2008). Case 2 captures traditional low-dimensional ($n > d$) ERM with convex losses or regularized losses as above with optimal regularization parameter $\mu = \mathcal{O}(1/\sqrt{N})$ (Shalev-Shwartz et al., 2009; Bartlett et al., 2006; Frostig et al., 2015).

Under (5), the suboptimality gap at given $x \in \mathcal{K}$ reads:[1]

$$F_P(x) - F_P(x^\star) \leq \mathcal{O}(V_N) + \left\{F(x) - F(\widehat{x})\right\}, \quad \text{w.h.p.} \tag{6}$$

Therefore, our ultimate goal will be computing ε-solutions x_ε of (P) of the order $\varepsilon = \mathcal{O}(V_N)$.

Statistical similarity: We are interested in studying problem (P) under statistical similarity of f_i's.

Assumption 4 (β-related f_i's). *The local functions f_i's are β-related: $\left\|\nabla^2 F(x) - \nabla^2 f_i(x)\right\|_2 \leq \beta$, for all $x \in \mathcal{K}$ and some $\beta \geq 0$.*

The interesting case is when $1+\beta/\mu \ll \kappa \triangleq Q/\mu$, where Q is the Lipschitz constant of ∇F on $\mathcal{K}$ (uniformly on $\mathcal{Z}$). Under standard assumptions on data distributions and learning model underlying the ERM-see, e.g., (Zhang & Xiao, 2015; Hendrikx et al., 2020b)–β is of the order $\beta = \mathcal{O}\left(1/\sqrt{n}\right)$, with high probability. In our analysis, when we target convergence to the statistical error, we will tacitly assume such dependence of β on the local sample size. Note that our bounds hold for general situations when Assumption 4 may hold due to some other reason besides statistical arguments.

2.2. Network setting

The network of agents is modeled as a fixed, undirected graph, $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \triangleq \{1, \ldots, m\}$ denotes the vertex set–the set of agents–while $\mathcal{E} \triangleq \{(i, j)\,|\, i, j \in \mathcal{V}\}$ represents the set of edges–the communication links; $(i, j) \in \mathcal{E}$ iff there exists a communication link between agent i and j. The following is a standard assumption on the connectivity.

Assumption 5 (On the network). *The graph $\mathcal{G}$ is connected.*

3. Algorithmic Design: DiRegINA

We aim at decentralizing the cubic regularization of the Newton method (Nesterov & Polyak, 2006) over undirected graphs. The main challenge in developing such an algorithm is to track and adapt a faithful estimates of the global gradient and Hessian matrix of F at each agent, without incurring in an unaffordable communication overhead while still guaranteeing convergence at fast rates. Our idea is to estimate locally the gradient ∇F via gradient-tracking (Xu et al., 2018; Di Lorenzo & Scutari, 2016) while the Hessian $\nabla^2 F$ is replaced by the local subsampled estimates $\nabla^2 f_i$ (statistical preconditioning). The algorithm, termed DiRegINA (Distributed Regularized Inexact Newton Algorithm), is formally introduced in Algorithm 1, and commented next.

[1] We point out that our results hold under (6), which can also be established using weaker conditions than (5), e.g., invoking stability arguments (Shalev-Shwartz et al., 2010).

Each agent maintains and updates iteratively a local copy $x_i \in \mathbb{R}^d$ of the global optimization variable x along with the auxiliary variable $s_i \in \mathbb{R}^d$, which estimates the gradient of the global objective F; x_i^ν (resp. s_i^ν) denotes the value of x_i (resp. s_i) at iteration $\nu \geq 0$. (`S.1`) is the optimization step wherein every agent i, given x_i^ν and s_i^ν, minimizes an inexact local second-order approximation of F, as defined in (7a). In this surrogate function, i) s_i^ν acts as an approximation of ∇F at x_i^ν, that is, $s_i^\nu \approx \nabla F(x_i^\nu)$; ii) in the quadratic term, $\nabla^2 f_i(x_i^\nu)$ plays the role of $\nabla^2 F(x_i^\nu)$ (due to statistical similarity, cf. Assumption 4) with $\tau_i I$ ensuring strong convexity of the objective; and iii) the last term is the cubic regularization as in the centralized method (Nesterov & Polyak, 2006). In (`S.2`), based upon exchange of the two vectors $x_i^{\nu+}$ and s_i^ν with their immediate neighbors, each agent updates the estimate $x_i^\nu \to x_i^{\nu+1}$ via the consensus step (7b) and $s_i^\nu \to s_i^{\nu+1}$ via the perturbed consensus (7c), which in fact tracks $\nabla F(x_i^\nu)$ (Xu et al., 2018; Di Lorenzo & Scutari, 2016). The weights $(W_K)_{i,j=1}^m$ in (7b)-(7c) are free design quantities and subject to the following conditions, where $\mathcal{P}_K$ denotes the set of polynomials with degree less than or equal than $K = 1, 2, \ldots$.

Assumption 6 (On the weight matrix W_K)**.** *The matrix $W_K = P^K(\overline{W})$, where $P_K \in \mathcal{P}_K$ with $P_K(1) = 1$, and $\overline{W} \triangleq (\bar{w}_{ij})_{i,j=1}^m$ is a reference matrix satisfying the following conditions:*

(a) *$\overline{W}$ has a sparsity pattern compliant with $\mathcal{G}$, that is*

i) $\bar{w}_{ii} > 0$, for all $i = 1, \ldots, m$;

ii) $\bar{w}_{ij} > 0$, if $(i,j) \in \mathcal{E}$; and $\bar{w}_{ij} = 0$ otherwise.

(b) *$\overline{W}$ is doubly stochastic, i.e., $1^\top \overline{W} = 1^\top$ and $\overline{W} 1 = 1$.*

Let $\rho_K \triangleq \lambda_{\max}(W_K - 11^\top/m)$ [$\lambda_{\max}(\bullet)$ denotes the largest eigenvalue of the matrix argument]

When $K = 1$, $W_K = \overline{W}$, that is, a single round of communication per iteration is performed. Several rules have been proposed in the literature for $\overline{W}$ to be compliant with Assumption 6, such as the Laplacian, the Metropolis-Hasting, and the maximum-degree weights rules; see, e.g., (Nedić et al., 2018) and references therein. When $K > 1$, K rounds of communications per iteration ν are employed. For instance, this can be performed using the same reference matrix $\overline{W}$ (satisfying Assumption 6) in each communication exchange, resulting in $W_K = \overline{W}^K$ and $\rho_K = \rho^K$, with $\rho = \lambda_{\max}(\overline{W} - 11^\top/m) < 1$. Faster information mixing can be obtained using suitably designed polynomials $P_K(\overline{W})$, such as Chebyshev (Wien, 2011; Scaman et al., 2017) or orthogonal (a.k.a. Jacobi) (Berthier et al., 2020) polynomials (notice that $P_K(1) = 1$ is to ensure the doubly stochasticity of W_K when $\overline{W}$ is doubly stochastic).

Although the minimization (7a) may look challenging, it is showed in (Nesterov & Polyak, 2006) that its computational complexity is of the same order as of finding the standard Newton step. Importantly, in our algorithm, these are local steps made without any communications between the nodes.

```
Algorithm 1 DiRegINA
Data: $x_i^0 \in \mathcal{K}$ and $s_i^0 = \nabla f_i(x_i^0)$, $\tau_i > 0$, $M_i > 0$, $\forall i$.
Iterate: $\nu = 1, 2, \ldots$
    [S.1] [Local Optimization] Each agent $i$ computes $x_i^{\nu+}$:
        $x_i^{\nu+} = \underset{y \in \mathcal{K}}{\text{argmin}}\ F(x_i^\nu) + \langle s_i^\nu, y - x_i^\nu \rangle$
        $+ \frac{1}{2} \left\langle \left[\nabla^2 f_i(x_i^\nu) + \tau_i I\right](y - x_i^\nu), y - x_i^\nu \right\rangle + \frac{M_i}{6} \|y - x_i^\nu\|^3.$    (7a)
    [S.2] [Local Communication] Each agent $i$ updates its local variables according to
        $x_i^{\nu+1} = \sum_{j=1}^m (W_K)_{i,j}\, x_j^{\nu+},$    (7b)
        $s_i^{\nu+1} = \sum_{j=1}^m (W_K)_{i,j}\, \left(s_j^\nu + \nabla f_j(x_j^{\nu+1}) - \nabla f_j(x_j^\nu)\right).$    (7c)
end
```

On the initialization: We will study convergence of Algorithm 1 under two sets of initialization for the x-variables, namely: i) random initialization and ii) statistically informed initialization. The latter is given by

$$x_i^0 = \sum_{j=1}^m (W_K)_{i,j} x_j^{-1}, \quad \text{with} \quad x_i^{-1} = \underset{x \in \mathcal{K}}{\text{argmin}}\, f_i(x). \tag{8}$$

This corresponds to a preliminary round of consensus on the local solutions x_i^{-1}. This second strategy takes advantage of the statistical similarity of f_i's to guarantee, under (5), an initial optimality gap of the order of: $p^0 \triangleq \frac{1}{m}\sum_{i=1}^m \left(F(x_i^0) - F(\widehat{x})\right) = \mathcal{O}(1/\sqrt{n})$. If we further assume $\mu_i > 0$, for all i, one can show that $p^0 = \mathcal{O}(1/n)$. This will be shown to significantly improve the convergence rate of the algorithm, at a negligible extra communication cost (but local computations).

4. Convergence Analysis

In this section, we study convergence of DiRegINA applied to convex (cf. Sec. 4.1) and strongly convex ERM (P), the latter with either $\beta < \mu$ (cf. Sec. 4.2) or $\beta \geq \mu > 0$ (cf. Sec. 4.3). Our complexity results are of two type: i) classical rate bounds targeting any arbitrary ERM suboptimality $\varepsilon > 0$; and ii) convergence rates to V_N-solutions of (P) (statistical error). Our complexity bounds are established in terms of the suboptimality gap:

$$p^{\nu} \triangleq \frac{1}{m}\sum_{i=1}^{m}\left(F(x_i^{\nu}) - F(\widehat{x})\right), \qquad (9)$$

where $\{x_i^{\nu}\}_{i=1}^m$ is the iterate generated by DiRegINA at iteration ν (iterations are counted as number of optimization steps `(S.1)`). Similarly to the centralized case (Nesterov & Polyak, 2006), our bounds also depend on the following distance of initial points x_i^0, $i = 1, \ldots, m$, from a given optimum $\widehat{x}$ of (P)

$$D \triangleq \max_{x_i \in \mathcal{K}, \forall i}\left\{\max_{i=1\ldots,m} ||x_i - \widehat{x}|| : \sum_{i=1}^{m} F(x_i) \leq \sum_{i=1}^{m} F(x_i^0)\right\}.$$

Note that $D < \infty$ (cf. Assumption 1).

For the sake of simplicity, in the rate bounds we hide universal constants and log factors independent on ε via $\widetilde{\mathcal{O}}$-notation; the exact expressions can be found in the supplementary material along with a detailed characterization of all the rate regions travelled by the algorithm.

4.1. Convex ERM (P)

Our first result pertains to convex F (and F_P).

Theorem 7. *Consider the ERM (P) under Assumptions 1, 3, and 4 over a graph $\mathcal{G}$ satisfying Assumption 5; and let $\{x_i^{\nu}\}_{i=1}^m$ be the sequence generated by DiRegINA under the following tuning: $M_i = L > 0$ and $\tau_i = 2\beta$, for all $i = 1, \ldots, m$; $W_K = P_K(\overline{W})$ (and $P_K(1) = 1$), where $\overline{W}$ is a given matrix satisfying Assumption 6 with $\rho = \lambda_{\max}(\overline{W} - 11^{\top}/m)$, and $K = \widetilde{\mathcal{O}}(\log(1/\varepsilon)/\sqrt{1-\rho})$, with $\varepsilon > 0$ being the target accuracy. Then, the total number of communications for DiRegINA to make $p^{\nu} \leq \varepsilon$ reads*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \left\{\sqrt{\frac{LD^3}{\varepsilon^{1+\alpha}}} + \frac{LD^3\beta}{\varepsilon^{1+\alpha/2}}\right\}\right), \qquad (10)$$

where $\alpha > 0$ is arbitrarily small. In particular, if the $\mathcal{G}$ is a star or fully-connected, $\rho = 0$ and $\alpha = 0$.

Proof. See Appendix D in the supplementary material. □

The rate expression (10) has an interesting interpretation. The multiplicative factor $1/\sqrt{1-\rho} > 1$ accounts for the rounds of communications per iteration (optimization steps) while the other two terms quantify the overall number of iterations to reach the desired accuracy ε. Note that the first of these two terms, $\mathcal{O}(\sqrt{LD^3/\varepsilon^{1+\alpha}})$, is "almost" identical to the rate of the centralized Newton method (with a slight difference definition of D; see (Nesterov & Polyak, 2006)) while the other one, $\mathcal{O}((LD^3\beta)/\varepsilon^{1+\alpha/2})$, is a byproduct of the discrepancy between local and global Hessian matrices. This shows a two-speed behavior of the algorithm, depending on the target accuracy $\varepsilon > 0$: 1) as long as $\varepsilon = \Omega(LD^3\beta^2)$, $\mathcal{O}((LD^3\beta^2)/\varepsilon)$ can be neglected and the algorithm exhibits almost centralized fast convergence (up to the network effect), $\mathcal{O}(\frac{1}{\sqrt{1-\rho}}\sqrt{LD^3/\varepsilon^{1+\alpha}})$; 2) on the other hand, for smaller (order of) ε, the rate is determined by the worst-term $\mathcal{O}(\frac{1}{\sqrt{1-\rho}}(LD^3\beta^2)/\varepsilon)$.

The interesting observation is that, in the setting above and under (5), (6) holds with $V_N = \mathcal{O}(1/\sqrt{N})$ and $\beta = \mathcal{O}(1/\sqrt{n})$. Hence, $\varepsilon = \Omega(LD^3\beta^2)$ is of the order of the statistical error V_N, as long as $m \leq n$, which is a reasonable condition. This together with Theorem 7 implies that fast rates (of the order of centralized ones) can be certified up to the statistical precision, as formalized next.

Corollary 8 (V_N-solution)**.** *Instate the setting of Theorem 7, and let $V_N = \mathcal{O}(1/\sqrt{N})$, $\beta = \mathcal{O}(1/\sqrt{n})$, and $m \leq n$. Then DiRegINA returns a V_N-solution of (P) in*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}} \cdot \sqrt{\frac{LD^3}{V_N^{1+\alpha}}}\right) \qquad (11)$$

communications.

4.2. Strongly-convex ERM (P) with $\beta < \mu$

We consider now the case of F μ-strongly convex and $\beta < \mu$. The complementary case $\beta \geq \mu$ is studied in Sec. 4.3.

Theorem 9. *Instate the setting of Theorem 7 with Assumption 1 replaced by Assumption 2 and $K = \widetilde{\mathcal{O}}(1/\sqrt{1-\rho})$; and further assume $\beta < \mu$. Then, the total number of communications for DiRegINA to make $p^{\nu} \leq \varepsilon$ reads*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{m^{\frac{1}{4}}\sqrt{\frac{LD}{\mu}} + \log\log\left[\frac{\mu^2}{\beta^2} \cdot \min\left(1, \frac{\beta^2\mu}{mL^2} \cdot \frac{1}{\varepsilon}\right)\right]\right.\right.$$
$$\left.\left. + \frac{\beta}{\mu}\log\left[\max\left(1, \frac{\beta^2\mu}{mL^2} \cdot \frac{1}{\varepsilon}\right)\right]\right\}\right). \qquad (12)$$

Proof. See Appendix E in the supplementary material. □

DiRegINA exhibits a different rate behavior, depending on the value of ϵ. We notice three "regions": 1) a first phase of the order of $\widetilde{\mathcal{O}}\big(m^{1/4}\sqrt{LD/\mu}\big)$ number of iterations; 2) the second region is of quadratic convergence, with rate of the order of $\log\log(1/\varepsilon)$; and finally 3) the region of linear convergence with rate $\widetilde{\mathcal{O}}\left(\beta/\mu\log(1/\varepsilon)\right)$. This last region is not present in the rate of the centralized cubic regularization of the Newton method and is due to the Hessians discrepancy. Clearly, for arbitrarily small $\varepsilon > 0$, (12) is dominated by the last term, resulting in a linear convergence. This linear rate is slightly worse than that of SONATA (Sun et al., 2019) in sight of first two terms in (12). This is because DiRegINA is an inexact (and thus more computationally efficient) method than (Sun et al., 2019). We remark that more favorable complexity estimates can be obtained when $L = 0$ (i.e., f_i's are quadratic)–we refer the reader to the supplementary material for details.

The algorithm does not enter in the last region if $\varepsilon = \Omega(\beta^2\mu/(mL^2))$. This means that faster rate can be guaranteed up to V_N-solutions, as stated next.

Corollary 10 (V_N-solution)**.** *Instate the setting of Theorem 9, and let $V_N = \mathcal{O}(1/N)$, $\beta = \mathcal{O}(1/\sqrt{n})$, $\mu = \mathcal{O}(1)$, and $m \le n$. DiRegINA returns a V_N-solution of (P) in*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{m^{1/4}\sqrt{\frac{LD}{\mu}} + \log\log\left(\frac{\mu^3}{mL^2V_N}\right)\right\}\right) \tag{13}$$

communications.

When the problem is ill-conditioned (i.e. $\mu \ll 1$) the first term $m^{1/4}\sqrt{LD/\mu}$ may dominate the $\log\log$ term in (13), unless n is extremely large (and thus V_N very small). This term is unavoidable–it is present also in the centralized instances of Newton-type methods–unless more refined function classes are considered, such as (generalized) self-concordant (Bach, 2010; Nesterov, 2018; Sun & Tran-Dinh, 2019). In the supplementary material, we present results for quadratic losses (cf. Appendix E.4). Here, we take another direction and show that the initialization strategy (8) is enough to get rid of the first phase.

Corollary 11 (V_N-solution + initialization)**.** *Instate the setting of Theorem 10 and further assume: $\mu_i = \Omega(1)$, for all $i = 1, \ldots m$, and $n = \Omega(L^2/\mu^3 \cdot m)$. DiRegINA , initialized with (8), returns a V_N-solution of (P) in*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{\log\log\left(\frac{\mu^3}{mL^2}\cdot\frac{1}{V_N}\right)\right\}\right) \tag{14}$$

communications.

Proof. See Appendix E.5 in the supporting material. □

4.3. Strongly-convex ERM (P) with $\beta \ge \mu$

We now consider the complementary case $\beta \ge \mu$. This is a common setting when F_P is convex and a regularizer is used in the ERM (P), making F μ-strongly convex; typically, $\mu = \mathcal{O}(1/\sqrt{N})$ while $\beta = \mathcal{O}(1/\sqrt{n})$.

Theorem 12. *Instate the setting of Theorem 9 with now $\mu \le \beta \le 1$. Then, the total number of communications for DiRegINA to make $p^\nu \le \varepsilon$ reads*

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\left\{\sqrt{\frac{LD}{\mu}}\left(1+m^{\frac{1}{4}}\sqrt{\frac{\beta}{\mu}}\right)+\frac{\beta}{\mu}\log\left(\frac{\beta^2\mu}{mL^2}\frac{1}{\varepsilon}\right)\right\}\right). \tag{15}$$

Proof. See Appendix F in the supplementary material. □

For arbitrary small $\varepsilon > 0$, the rate (15) is dominated by the linear term. When we target V_N-solutions, in this setting $V_N = \mathcal{O}(1/\sqrt{N})$, $\mu = \mathcal{O}(V_N)$ (as for the regularized ERM setting), and $\beta = \mathcal{O}(1/\sqrt{n})$, (15) becomes

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\cdot m^{1/2}\cdot\sqrt{\frac{LD}{V_N}}\right). \tag{16}$$

Note that this rate is of the same order of the one achieved in the convex setting (with no regularization)–see Corollary 8. If the functions f_i are quadratic, the rate, as expected, improves and reads (see supporting material, Appendix G)

$$\widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{1-\rho}}\cdot m^{1/2}\cdot\log\left(\frac{1}{V_N}\right)\right).$$

Note that, on star networks ($\rho = 0$), this rate improves on that of DANE (Shamir et al., 2014).

5. Experiments

In this section we test numerically our theoretical findings on two classes of problems over meshed networks: 1) ridge regression and 2) logistic regression. Other experiments can be found in the supplementary material (cf. Sec. A).

The network graph is generated using an Erdős-Rényi model $G(m,p)$, with $m = 30$ nodes and different values of p to span different level of connectivity.

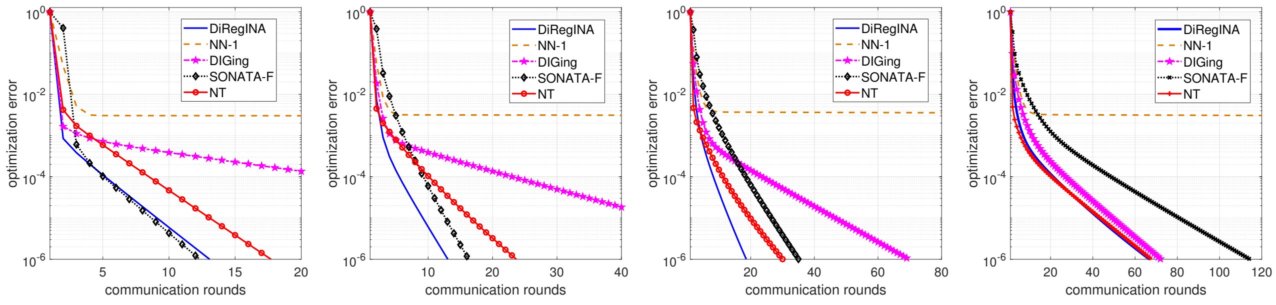
We compare DiRegINA with the following methods:

• *Distributed (first-order) method with gradient tracking:* we consider SONATA (Sun et al., 2019) and DIGing (Nedic et al., 2017); both build on the idea of gradient tracking, with the former applicable also to constrained problems. For the SONATA algorithm, we will simulate two instances, namely: SONATA-L (L stands for linearization) and SONATA-F (F stands for full); the former uses only first-order information in the agents' local updates (as DGing) while the latter exploits functions' similarity by employing local mirror-descent-based optimization.

• *Distributed accelerated first-order methods:* we consider APAPC (Kovalev et al., 2020) and SSDA (Scaman et al., 2017), which employ Nesterov acceleration on the local optimization steps–with the former using primal gradients while the latter requiring gradients of the conjugate functions–and Chebyshev acceleration on the consensus steps. These schemes do not leverage any similarity among the local agents' functions.

• *Distributed second-order methods:* We implement i) Network Newton-K (NN-K) (Mokhtari et al., 2016b) with $K = 1$ so that it has the same communication cost per iteration of DiRegINA ; ii) SONATA-F (Sun et al., 2019), which is a mirror descent-type distributed scheme wherein agents need to solve *exactly* a strongly convex optimization problem; and iii) Newton Tracking (NT) (Jiaojiao et al., 2020), which has been shown the outperform the majority of distributed second-order methods.

All the algorithms are coded in MATLAB R2019a, running on a computer with Intel(R) Core(TM) i7-8650U CPU@1.90GHz, 16.0 GB of RAM, and 64-bit Windows 10.



(a) (b) (c) (d)

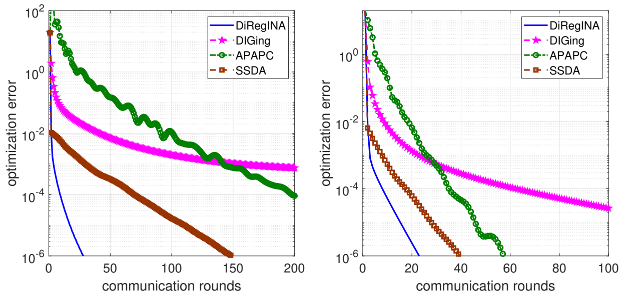
Figure 1. Distributed ridge regression: (a) star-topology; and Erdős-Rényi graph with (b) $\rho = 0.20$, (c) $\rho = 0.41$, (d) $\rho = 0.69$.

5.1. Distributed Ridge Regression

We train ridge regression, LIBSVM, scaled mg dataset (Flake & Lawrence, 2002), which is an instance of (P) with $f_i(x) = (1/2n)\left\|A_i x - b_i\right\|^2 + \frac{\lambda}{2}\left\|x\right\|^2$ and $\mathcal{K} = \mathbb{R}^d$, with $d = 6$. We set $\lambda = 1/\sqrt{N} = 0.0269$; we estimate $\beta = 0.1457$ and $\mu = 0.0929$. The graph parameter $p = 0.6, 0.33, 0.28$, resulting in the connectivity values $\rho \approx 0.20, 0.41, 0.70$, respectively. We compared DiRegINA, NN-1, DIGing, SONATA-F and NT, all initialized from the same identical random point. The coefficients of the matrix $\overline{W}$ are chosen according to the Metropolis–Hastings rule (Xiao et al., 2007). The free parameters of the algorithm are tuned manually; specifically: DiRegINA, $\tau = 2\beta$, $M = 1e-3$, and $K = 1$; NN-1, $\alpha = 1e-3$ and $\epsilon = 1$; DIGing, stepsize equal to 0.5; SONATA-F, $\tau = 0.27$; NT, $\epsilon = 0.08$ and $\alpha = 0.1$. This tuning corresponds to the best practical performance we observed.

In Fig. 1, we plot the function residual p^ν defined in (9) versus the communication rounds in the four aforementioned network settings. DiRegINA demonstrates good performance over first-order methods, and compares favorably also with SONATA-F (which has higher computational cost). Note the change of rate, as predicted by our theory, with linear rate in the last stage. NN-1 is not competitive while NT in some settings is comparable with DiRegINA , but we observed to be more sensitive to the tuning.

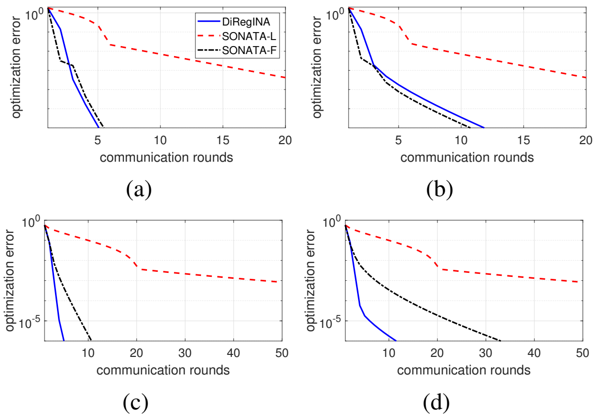
The second experiment aims at comparing DiRegINA with the distributed accelerated methods APAPC (Kovalev et al., 2020) and SSDA (Scaman et al., 2017) (DIGing is used as benchmark of first-order non-accelerated schemes). We tested these schemes on two instances of the Ridge regression problem using synthetic data, corresponding to $\beta/\mu \gg \sqrt{\kappa}$ and $\beta/\mu \approx \sqrt{\kappa}$. Recall that SSDA and APAPC converge linearly at a rate proportional to $\sqrt{\kappa}$ while the convergence rate of DiRegINA depends (up to log factors) on β/μ. The problem data are generated as follows: the ground truth $x^* \in \mathbb{R}^d$ is a random vector, $x^* \sim \mathcal{N}(\mathbf{0}, I)$, with $d = 40$; samples $b_i \triangleq (b_i^{(j)})_{j=1}^n$, with $n = 50$, are generated according to the linear model $b_i^{(j)} = a_i^{(j)\top} x^* + \epsilon_i^{(j)}$ where $\epsilon_i^{(j)} \sim \mathcal{N}(0, 1e-4)$. To obtain controlled values for β, $A_i \triangleq (a_i^{(j)})_{j=1}^n$ are constructed as follows: we first generate n i.i.d samples $A_1 \triangleq (a_1^{(j)})_{j=1}^n$, with rows drawn from $\mathcal{N}(\mathbf{0}, I)$; then, we set each $A_i = A_1 + E_k$, where E_k in a random matrix with rows drawn from $\mathcal{N}(\mathbf{0}, \sigma I)$. The choices of σ are considered resulting in two different values of β, namely: $\sigma = 1/(dn)$ and $\sigma = 7.5/(dn)$, resulting in $\beta = 0.31$ and $\beta = 4.08$, respectively. The values of the condition number read $\kappa = 123.21$ and $\kappa = 1.19e3$, respectively. The network is simulated as the Erdős-Rényi graph with $p = 0.28$, resulting in $\rho \approx 0.7$; the number of agents is $m = 30$. The tuning of DiRegINA and DIGing is the same as in Fig. 1 while APAPC and SSDA are manually tuned for best practical performance.



(a) (b)

Figure 2. Distributed ridge regression. Synthetic data on Erdős-Rényi graph with $\rho = 0.7$: a) $\beta/\mu = 158.1$, $\sqrt{\kappa} = 34.55$; b) $\beta/\mu = 11.974$, $\sqrt{\kappa} = 11.1$.

In Fig. 2, we plot the function residual p^ν defined in (9) versus the communication rounds; the two panels refer to two different values of $(\beta/\mu, \sqrt{\kappa})$. The figures show that even when β/μ is larger than $\sqrt{\kappa}$, DiRegINA outperforms the accelerated first order methods; roughly, it is from two to five time faster than the best simulated first order method.



(a) (b) (c) (d)

Figure 3. Distributed logistic regression: 1) a4a dataset on Erdős-Rényi graph with (a) $\rho = 0.367$ (b) $\rho = 0.757$; 2) Synthetic data on Erdős-Rényi graph with (c) $\rho = 0.367$ (d) $\rho = 0.757$.

5.2. Distributed Logistic Regression

We train logistic regression models, regularized by the ℓ_2-ball constraint (with radius 1). The problem is an instance of (P), with each $f_i(x) = -(1/n)\sum_{j=1}^{n}[\xi_i^{(j)}\ln(z_i^{(j)}) + (1-\xi_i^{(j)})\ln(1-z_i^{(j)})]$, where $z_i^{(j)} \triangleq 1/(1+e^{-\langle a_i^{(j)},x\rangle})$ and binary class labels $\xi_i^{(j)} \in \{0,1\}$ and vectors $a_i^{(j)}$, $i = 1,\ldots m$ and $j = 1,\ldots,n$ are determined by the data set. We considered the LIBSVM a4a ($N = 4,781$, $d = 123$) and synthetic data ($N = 900$, $d = 150$). The latter are generated as follows: a random ground truth $x^* \sim \mathcal{N}(\mathbf{0}, I)$, i.i.d. sample $\{a_i^{(j)}\}_{i,j}$, and $\{\xi_i^{(j)}\}_{i,j}$ are generated according to the binary model $\xi_i^{(j)} = 1$ if $\langle a_i^{(j)}, x^*\rangle \geq 0$ and $\xi_i^{(j)} = 0$ otherwise. We consider Erdős-Rényi network models with connectivity $\rho = 0.367$ and $\rho = 0.757$.

We compare DiRegINA with SONATA-F and SONATA-L, since they are the only two algorithms in the list that can handle constrained problems. We report results obtained under the following tuning: (i) both SONATA variants, $\alpha = 0.1$; and (ii) DiRegINA , $M = 1$ and $\tau_i = 1e-3$. The coefficients of the matrix $\overline{W}$ are chosen according to the Metropolis–Hastings rule (Xiao et al., 2007).

In Fig. 3, we plot the function residual p^ν defined in (9) versus the communication rounds, in the different mentioned network settings. On real data [panels (a)-(b)], DiRegINA and SONATA-F performs equally well, outperforming SONATA-L (first-order method). When tested on the synthetic problem [panel (c)-(d)] with less local samples n and larger dimension d, DiRegINA shows a consistently faster rate, while SONATA-F slows down on less connected networks. Notice also the two-phase rate of DiRegINA , as predicted by our theory: an initial superlinear rate up to (approximately) the statistical precision, followed by a linear one for high accuracy.

6. Conclusions

We proposed the first second-order distributed algorithm for convex and strongly convex problems over meshed networks with *global* communication complexity bounds which, up to the network dependent factor $\widetilde{\mathcal{O}}(1/\sqrt{1-\rho})$, (almost) match the iteration complexity of centralized second-order method (Nesterov & Polyak, 2006) in the regime when the desired accuracy is moderate. We showed that this regime is reasonable when one considers ERM problems for which there is no need to optimize beyond the statistical error. Importantly, our method avoids expensive communications of Hessians over the network and keeps the amount of information sent in each communication round similar to first-order methods.

This paper is just a starting point towards a theory of second-order methods with performance guarantees on meshed networks under statistical similarity; many questions remain open. An obvious one is incorporating acceleration to improve communication complexity bounds under statistical similarity. A first attempt towards this goal is the follow-up work (Agafonov et al., 2021), where an accelerated second-order method exploiting statistical similarity has been analyzed for master/workers architectures. The extension to arbitrary graphs remains an open problem. Second, our main goal here has been decreasing communications, which does not guarantee optimal oracle (computational) complexity–this is because we did not take advantage of the finite-sum structure of the *local* optimization problems. Stochastic optimization algorithms equipped with Variance Reduction (VR) techniques have been proved to be quite effective to obtain cheaper iterations while preserving fast convergence (Johnson & Zhang, 2013; Hendrikx et al., 2020a). However, these methods do not exploit any statistical similarity, resulting in less favorable communication complexity whenever $\beta/\mu \ll Q/\mu$. It would be then interesting to investigate whether VR techniques can improve both communication and oracle complexity when statistical similarity is explicitly employed in the algorithmic design.

Acknowledgements

The work of A. Daneshmand and G. Scutari was supported by the USA NSF Grant CIF 1719205, the ARO Grant No. W911NF1810238, and the ONR Grant No. 1317827. The work of P. Dvurechensky and A. Gasnikov was prepared within the framework of the HSE University Basic Research Program and is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) No. 075-00337-20-03, project No. 0714-2020-0005.

References

Agafonov, A., Dvurechensky, P., Scutari, G., Gasnikov, A., Kamzolov, D., Lukashevich, A., and Daneshmand,

A. An accelerated second-order method for distributed stochastic optimization. *arXiv:2103.14392*, 2021.

Arjevani, Y. and Shamir, O. Communication complexity of distributed convex learning and optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, volume 1, pp. 1756–1764, December 2015.

Bach, F. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414., 2010.

Bartlett, P. L., Jordan, M. I., and McAulffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Bekkerman, R., Bilenko, M., and Langford, J. *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press, 2011.

Berthier, R., Bach, F., and Gaillard, P. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM J. on Mathematics of Data Science*, 1: 24–47, 2020.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Bousquet, O. *Concentration inequalities and empirical processes theory applied to the analysis of learning algorithms*. PhD thesis, Ecole Polytechnique: Department of Applied Mathematics Paris, France, 2002.

Di Lorenzo, P. and Scutari, G. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, June 2016.

Dvurechensky, P., Kamzolov, D., Lukashevich, A., Lee, S., Ordentlich, E., Uribe, C. A., and Gasnikov, A. Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization. *arXiv:2102.08246*, 2021.

Eisen, M., Mokhtari, A., and Ribeiro, A. A primal-dual quasi-newton method for exact consensus optimization. *IEEE Transactions on Signal Processing*, 67(23):5983–5997, 2019.

Fan, J., Guo, Y., and Wang, K. Communication-efficient accurate statistical estimation. *arXiv:1906.04870*, 2019.

Flake, G. W. and Lawrence, S. Efficient SVM regression training with SMO. *Machine Learning*, 46(1):271–290, 2002.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory (COLT)*, pp. 728–763, 2015.

Gorbunov, E., Rogozin, A., Beznosikov, A., Dvinskikh, D., and Gasnikov, A. Recent theoretical advances in decentralized distributed convex optimization. *arXiv:2011.13259*, 2020.

Hendrikx, H., Bach, F., and Massoulie, L. An optimal algorithm for decentralized finite sum optimization. *arXiv:2005.10675*, 2020a.

Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., and Massoulie, L. Statistically preconditioned accelerated gradient method for distributed optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 4203–4227, 13–18 Jul 2020b.

Jadbabaie, A., Ozdaglar, A., and Zargham, M. A distributed newton method for network optimization. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 2736–2741, 2009. doi: 10.1109/CDC.2009.5400289.

Jahani, M., He, X., Ma, C., Mokhtari, A., Mudigere, D., Ribeiro, A., and Takac, M. Efficient distributed hessian free algorithm for large-scale empirical risk minimization via accumulating sample strategy. In *Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 2634–2644, 2020.

Jakovetić, D., Xavier, J., and Moura, J. M. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

Jiaojiao, Z., Ling, Q., and So, A. A newton tracking algorithm with exact linear convergence rate for decentralized consensus optimization. In *59th IEEE Conference on Decision and Control (CDC)*, 2020.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

Kovalev, D., Salim, A., and Richtárik, P. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Lan, G., Lee, S., and Zhou, Y. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pp. 1–48, 2017.

Li, B., Cen, S., Chen, Y., and Chi, Y. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *arXiv:1909.05844v3*, 2019.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu., J. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 5330–5340, 2017.

Ma, C. and Takac, M. Distributed inexact damped newton method: Data partitioning and work-balancing. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Mokhtari, A., Daneshmand, H., Lucchi, A., Hofmann, T., and Ribeiro, A. Adaptive newton method for empirical risk minimization to statistical accuracy. In *Proceedings of the Advances in Neural Information Processing Systems*, pp. 4062–4070, 2016a.

Mokhtari, A., Ling, Q., and Ribeiro, A. Network newton distributed optimization methods. *IEEE Transactions on Signal Processing*, 65(1):146–161, 2016b.

Mokhtari, A., Shi, W., Ling, Q., and Ribeiro, A. DQM: Decentralized quadratically approximated alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 64(19):5158–5173, 2016c.

Mokhtari, A., Shi, W., Ling, Q., and Ribeiro, A. A decentralized second-order method with exact linear convergence rate for consensus optimization. *IEEE Transactions on Signal Processing*, 2(4):507–522, 2016d.

Mokhtari, A., Ling, Q., and Ribeiro, A. Network newton distributed optimization methods. *IEEE Transactions on Signal Processing*, 65:146–161, 2017.

Nedic, A., Olshevsky, A., and Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4): 2597–2633, 2017.

Nedić, A., Olshevsky, A., and Rabbat, M. G. Network topology and communication – computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106 (5):953–976, 2018.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nesterov, Y. and Polyak, B. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108:177–205, 2006.

Reddi, S. J., Konečnỳ, J., Richtárik, P., Póczós, B., and Smola, A. Aide: Fast and communication efficient distributed optimization. *arXiv:1608.06879*, 2016.

Rogozin, A., Lukoshkin, V., Gasnikov, A., Kovalev, D., and Shulgin, E. Towards accelerated rates for distributed optimization over time-varying networks. *arXiv:2009.11069*, 2020.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3027–3036, 2017.

Shai, S.-S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorihtms*. Cambridge University Press, 2014.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, Montreal, Canada, June 18-21 2009.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

Shamir, O., Srebro, N., and Zhang, T. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31st International Conference on Machine Learning (PMLR)*, volume 32, pp. 1000–1008, 2014.

Shi, W., Ling, Q., Wu, G., and Yin, W. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Soori, S., Mishchenko, K., Mokhtari, A., Dehnavi, M. M., and Gurbuzbalaban, M. Dave-qn: A distributed averaged quasi-newton method with local superlinear convergence rate. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 1965–1976, 2020.

Sridharan, K., Shalev-Shwartz, S., and Srebro, N. Fast rates for regularized objectives. *Advances in neural information processing systems*, 21:1545–1552, 2008.

Sun, T. and Tran-Dinh, Q. Generalized self-concordant functions: a recipe for newton-type methods. *Mathematical Programming*, 178:145–213, 2019.

Sun, Y., Daneshmand, A., and Scutari, G. Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation. *arXiv:1905.02637*, 2019.

Tutunov, R., Bou-Ammar, H., and Jadbabaie, A. Distributed newton method for large-scale consensus optimization. *IEEE Transactions on Automatic Control*, 64(10):3983–3994, 2019. doi: 10.1109/TAC.2019.2907711.

Uribe, C. A. and Jadbabaie, A. A distributed cubic-regularized newton method for smooth convex optimization over networks. *arXiv:2007.03562*, 2020a.

Uribe, C. A. and Jadbabaie, A. A distributed cubic-regularized newton method for smooth convex optimization over networks, 2020b.

Uribe, C. A., Lee, S., Gasnikov, A., and Nedić, A. A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pp. 1–40, 2020.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 2013.

Wang, S., Roosta-Khorasani, F., Xu, P., and Mahoney, M. W. Giant: Globally improved approximate newton method for distributed optimization. In *Proceedings of the 32nd 32nd International Conference on Neural Information Processing Systems*, volume 37, pp. 2338–2348, 2018.

Wei, E., Ozdaglar, A., and Jadbabaie, A. A distributed newton method for network utility maximization—part ii: Convergence. *IEEE Transactions on Automatic Control*, 58(9):2176–2188, 2013. doi: 10.1109/TAC.2013.2253223.

Wien, A. *Iterative solution of large linear systems*. Lecture Notes, TU Wien, 2011.

Xiao, L., Boyd, S., and Kim, S.-J. Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1):33–46, 2007.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. Convergence of Asynchronous Distributed Gradient Methods Over Stochastic Networks. *IEEE Transactions on Automatic Control*, 63(2):434–448, 2018.

Yuan, X.-T. and Li, P. On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond. *arXiv:1908.02246*, 2019.

Zhang, J., You, K., and Basar, T. Distributed adaptive newton methods with globally superlinear convergence. *arXiv:2002.07378*, 2020.

Zhang, Y. and Xiao, L. Disco: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, volume 37, pp. 362–370, 2015.