Optimal Data Splitting in Distributed Optimization for Machine Learning*

Daniil Medyakov[1], Gleb Molodtsov[1], Aleksandr Beznosikov[1,2], and Alexander Gasnikov[1,2]

[1] Moscow Institute of Physics and Technology, Moscow, Russia,
[2] Institute for Information Transmission Problems, Moscow, Russia

Abstract. The distributed optimization problem has become increasingly relevant recently. It has a lot of advantages such as processing a large amount of data in less time compared to non-distributed methods. However, most distributed approaches suffer from a significant bottleneck – the cost of communications. Therefore, a large amount of research has recently been directed at solving this problem. One such approach uses local data similarity. In particular, there exists an algorithm provably optimally exploiting the similarity property. But this result, as well as results from other works solve the communication bottleneck by focusing only on the fact that communication is significantly more expensive than local computing and does not take into account the various capacities of network devices and the different relationship between communication time and local computing expenses. We consider this setup and the objective of this study is to achieve an optimal ratio of distributed data between the server and local machines for any costs of communications and local computations. The running times of the network are compared between uniform and optimal distributions. The superior theoretical performance of our solutions is experimentally validated.

1 Introduction

1.1 Distributed optimization

We consider optimization problems of the following form:

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{1}$$

where $x \in \mathbb{R}^d$ collects parameters of a statistical model to be trained, n is a number of devices/nodes, and f_i is an emperical risk of the devices i, i.e., $f_i(x) = \frac{1}{b_i} \sum_{j=1}^{b_i} l(x, z_i^j)$ with $z_i^1, \ldots, z_i^{b_i}$ is a set of b_i samples owned by the i-th device and $l(x, z_i^j)$ measures mismatch between the parameter x and the label

*The research of A. Beznosikov was supported by Russian Science Foundation (project No. 23-11-00229).

of the sample z_i^j. This is a direct formulation of the distributed optimization problem. Nowadays, there are some reasons to consider this.

To achieve the best results in modern machine learning optimization problems, researchers and practitioners face various challenges. Dealing with modern machine learning models remains an extremely challenging task, primarily because models are trained on increasingly large datasets. Having more data in the training sample increases the robustness and generalizability of the derived model. In this case, the data is typically processed using a network of devices, i.e., collected in a distributed manner and stored in edge nodes of the network, such as in classical clustering [20] and federated [9, 11, 7] learning.

Several solution methods have been proposed to solve (1). The prototype approach involves interleaving edge devices calculations (nodes $i = 2, \ldots, n$) with communications to and from the server ($i = 1$), which maintains and updates the authoritative copy of the optimization variables, eventually producing the final solution estimate. In distributed learning of complex models, the communication overhead between devices in the network often becomes a bottleneck. Such a problem makes it necessary to develop more efficient distributed learning methods, some of which have been described in [9, 4, 17, 5].

1.2 Distributed optimization under similarity

It is trendy today in machine learning to use momentum-based methods. One of the methods for solving a distributed optimization problem is the application of Nesterov acceleration [13], which is an optimal method for smooth non-distributed deterministic optimization problems. This method can be applicable to distributed networks as follows. At each iteration we calculate the gradient locally and send the results to the server. The server average the obtained gradients and make a method step. Then the number of communications is equal to the number of iterations. In this case, we obtain optimal estimates for local computations – $\sqrt{\kappa}, \kappa = {}^{L}/_{\mu}$, where L and μ are constants of smoothness and strong convexity of target function f. In case κ is small, this approach is acceptable. However, for ill-conditioned functions with a large κ, the polynomial dependence on κ may be unsatisfactory, due to the high cost of communications. This is often the case for many empirical risk minimization (ERM) problems where the optimal regularization parameter for test predictive performance is very small.

To further improve communication complexity, we can exploit the additional structure typically found in ERM problems, known as data similarity [1, 16, 12]. One can define it as the difference of function gradients, i.e., $\|\nabla f_i(x) - \nabla f_j(x)\| < \delta \ \ \forall x$. But this approach is not "natural", since if the problem is not bounded, such a δ cannot exist. Let us consider for example a quadratic problem: $\nexists \ \delta : \|(A_i - A_j)x\| \leq \delta$ if $x \to \infty$. Therefore, we focus on a different setting, namely on a Hessian similarity. Specifically, for all x in a suitable domain of interest and all $i \neq j; \ i, j \in \{1, \ldots, n\}$, the difference between the Hessian matrices of local losses, denoted by $\|\nabla^2 f_i(x) - \nabla^2 f_j(x)\|$, is bounded by δ, where $\delta > 0$ measures the degree of similarity. Under this assumption, we can estimate $\delta \sim \mathcal{O}({}^{1}/\sqrt{N})$, where N is the sample size per device [1]. This setting was investigated for the first time in [16]. After that, lower bounds were proved for this problem in [1], where communication costs are proportional to $\sqrt{\delta/\mu}$. Then for a long time researchers tried to find methods that would reach these estimates. In particular, algorithms such as [19, 18, 15, 6, 2] was obtained. In 2022, it was possible to find the optimal method which is described in [10].

1.3 Various communication costs and local computations

In these works, the authors made an assumption that communication costs significantly more than for local computations. Moreover, in general, works on distributed optimization, not only about the Hessian similarity, made this assumption. We look at this question from a different angle, move away from fixed big communications and make 2 **assumptions**:

1. The devices in the network have different capacities, i.e., they perform local computations of the same amount of data for different times.
2. The ratio of communication costs to local computation time is a variable value that can be either $\ll 1$, $\gg 1$, or even ~ 1.

Under such assumptions, we need a new approach to the distributed optimization problem based on the already obtained optimal algorithms. This leads us to the research question of this paper:

> *Can we find such a distribution of data among the devices in the network to reduce the actual running time of the optimal algorithm* [10] *for any communication costs and local computations?*

In practice, networks can run for long periods of time and as a consequence, noise can occur. In other words, communication costs and device capacities are not constant values. In that way, we make one more **assumption**:

3. We put communication costs and device capacities as random variables, start the network operation for a long time and measure their expectation and variance. Due to the fact that the distribution of data to devices depends on the constant communication time and device power, in reality, the optimal distribution is different on account of noise.

Therefore, this assumption raises one more research question of measuring the variance of the program running time under the optimal data distribution.

1.4 Contributions

In general, we can summarize our contribution as follows:

- **Generalization of the computation model.** We build a general model for computing time in networks under distributed optimization. The model is based on the optimal algorithm [10] and takes into account the difference in capacities of devices, in various communication costs.

- **Comprehensive analysis.** We pay special attention to the particular cases and obtain results for them. We consider the case where communications are too expensive and the case of inexpensive communications (not so expensive that the communication takes longer than processing all data by just one device). Moreover, we obtain results not only taking into account the difference in time costs, but we also consider different estimates on δ.
- **Different techniques for obtaining a solution.** We use different techniques: Cardano's formula, upper estimates in particular cases, finding the zero of the function using the simplest numerical methods.
- **Decision error due to noise.** Under the third assumption, we present the theoretical error of the program running time under communication and local capacities noise.
- **Experiments.** We also conduct experiments confirming that with the obtained distribution it takes less time to solve the distributed problem. Besides, we also make appropriate experiments with noise in the network.

2 Problem Statement

Let us stay under just the first two assumptions from Section 1.3 for now. To achieve lower communication and local gradient complexity, we can refer to Algorithm 1 from [10]. For this purpose, the function need to be represented as a sum of a smooth convex function f_1 and a smooth potentially non-convex function $f - f_1$. Then the algorithm is rewritten in the following form:

Algorithm 1 Accelerated Extragradient

```
2
1: Input: $x^0 = x_f^0 \in \mathbb{R}^d$
2: Parameters: $\tau \in (0,1), \eta, \theta, \alpha > 0, K \in \{1, 2, \ldots\}$
3: for $k = 0, 1, 2, \ldots, K-1$ do
4:     $x_g^k = \tau x^k + (1-\tau) x_f^k$
5:     $x_f^{k+1} \approx \arg\min_{x \in \mathbb{R}^d} [A_\theta^k(x) := \langle \nabla (f - f_1)(x_g^k), x - x_g^k \rangle + \frac{1}{2\theta} \|x - x_g^k\|^2 + f_1(x)]$
6:     $x^{k+1} = x^k + \eta\alpha(x_f^{k+1} - x^k) - \eta \nabla f(x_f^{k+1})$
7: end for
8: Output: $x^K$
```

We analyze the work of this algorithm, namely, find out how many operations this algorithm performs per iteration. In line 5, when solving the arg min subproblem, one local computation is performed on devices to compute $f_i(x_g^k)$, followed by one communication to transmit these results, and additional computations on the first device to find the solution x_f^{k+1}. Then in line 6 there is one local computation on all devices, and one communication. We obtain an expression for the total running time of the algorithm. Let us introduce the following notations: τ_i – time of one local computation on the i-th device, K – number of iterations, τ_{comm} – time of one communication, k_{some} – additional computation of the first/central node, n – number of nodes in the network. Taking this into account, the total running time of the algorithm can be written as:

$$T_{sum} = 2 \cdot \max(\tau_1, \tau_2, \ldots, \tau_n) \cdot K + 2 \cdot K \cdot \tau_{comm} + \tau_1 \cdot k_{some}.$$

Our task is to minimize the time T_{sum}. In view of the statement (1) and the form of functions f_i let us represent the time τ_i as $\tau_i = \tau_i^{loc} \cdot b_i$, where τ_i^{loc} is capacity, i.e., the time spent by the i-th device to process a unit of information submitted to its input, and b_i is the size of dataset submitted to the i-th device. All b_i satisfy the following constraints: $\sum\limits_{i=1}^{n} b_i = N$, where N is the size of the whole dataset, $\delta = \frac{L}{\sqrt{b_i}}$ or $\delta = \frac{L}{b_i}$ [6]. Finally, we obtain the following optimization problem:

$$\min_{\sum\limits_{i=1}^{n} b_i = N; \delta = \frac{L}{{b_1}^{\gamma}}} [2 \cdot \max(\tau_1^{loc} \cdot b_1, \ldots, \tau_n^{loc} \cdot b_n) \cdot K + 2 \cdot K \cdot \tau_{comm} + \tau_1 \cdot k_{some}],\ \gamma \in \{\frac{1}{2}, 1\}. \tag{2}$$

3 How to solve (2)

3.1 The primary problem of minimization

In the work [10] the estimates of K and k_{some} are presented, namely: $2 \cdot K = \mathcal{O}(\max\{1, \sqrt{\frac{\delta}{\mu}}\} \log \frac{1}{\varepsilon})$, $k_{some} = \mathcal{O}(\max\{1, \sqrt{\frac{L}{\delta}}, \sqrt{\frac{\delta}{\mu}}, \sqrt{\frac{L}{\mu}}\} \log \frac{1}{\varepsilon})$.

Thus, (2) is reduced to:

$$\begin{aligned} \min_{\sum\limits_{i=1}^{n} b_i = N; \delta = \frac{L}{{b_1}^{\gamma}}} & [(\max(\tau_1^{loc} \cdot b_1, \ldots, \tau_n^{loc} \cdot b_n) + \tau_{comm}) \cdot \mathcal{O}(\max\{1, \sqrt{\frac{\delta}{\mu}} \log \frac{1}{\varepsilon}\}) \\ & + \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\max\{1, \sqrt{\frac{L}{\delta}}, \sqrt{\frac{\delta}{\mu}}, \sqrt{\frac{L}{\mu}}\} \log \frac{1}{\varepsilon})],\ \gamma \in \{\frac{1}{2}, 1\}. \end{aligned} \tag{3}$$

3.2 Auxiliary problem

Consider an auxiliary problem:

$$\min_{\sum\limits_{i=2}^{n} b_i = N} [\max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)]. \tag{4}$$

Lemma 1. *The solution of problem* (4) *is* $\overrightarrow{b} = (b_2, b_3, \ldots, b_n)^T$ *satisfying* $\tau_2^{loc} \cdot b_2 = \tau_3^{loc} \cdot b_3 = \ldots = \tau_n^{loc} \cdot b_n$.

Proof. *Without loss of generality, let us assume fixed values for* $\tau_2^{loc} \leq \tau_3^{loc} \leq \ldots \leq \tau_n^{loc}$. *Then let us arbitrarily choose* $b_2 \geq b_3 \geq \ldots \geq b_n$. *This is indeed the case, otherwise we would have a situation where* $\exists\ i \neq j :\ i, j \in \{2, \ldots, n\} : \max(\tau_i^{loc} \cdot b_i, \tau_j^{loc} \cdot b_j) > \max(\tau_i^{loc} \cdot b_j, \tau_j^{loc} \cdot b_i)$, *and therefore the distribution would be suboptimal.*

Our goal is to minimize the function $g(\overrightarrow{b}) = \max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)$. *Suppose that there exists a distribution such that* $\exists i \in \{2, \ldots, n\} : g(\overrightarrow{b}^0) = \tau_i^{loc} \cdot b_i^0$ *is the minimum, and* $\forall j : j \geq 2, j \neq i \hookrightarrow \tau_i^{loc} \cdot b_i^0 > \tau_j^{loc} \cdot b_j^0$. *It follows that* $b_i^0 > \frac{\tau_j^{loc}}{\tau_i^{loc}} b_j^0 > \frac{\tau_{j_1}^{loc}}{\tau_i^{loc}} b_{j_1}^0 > \ldots > \frac{\tau_{j_k}^{loc}}{\tau_i^{loc}} b_{j_k}^0$. *Then, considering* $\sum\limits_{i=2}^{n} b_i = N \hookrightarrow b_i^0 + \frac{\tau_j^{loc}}{\tau_i^{loc}} b_j^0 + \frac{\tau_{j_1}^{loc}}{\tau_i^{loc}} b_{j_1}^0 + \ldots + \frac{\tau_{j_k}^{loc}}{\tau_i^{loc}} b_{j_k}^0 > N$, *we obtain* $b_i^0 > N(1 + \tau_i^{loc} \sum\limits_{j=2, j\neq i}^{n} \frac{1}{\tau_j^{loc}})^{-1}$.

Next, let us consider $b_i = N(1 + \tau_i^{loc} \sum\limits_{j=2, j\neq i}^{n} \frac{1}{\tau_j^{loc}})^{-1}, \quad b_j = \frac{\tau_i^{loc}}{\tau_j^{loc}} \cdot b_i \ \ \forall j \in \{2, \ldots, n\}$. *This distribution yields a minimum of* $g(\overrightarrow{b}) = \tau_i^{loc} \cdot b_i = \tau_j^{loc} \cdot b_j \ \ \forall j \in \{2, \ldots, n\}$, *and* $g(\overrightarrow{b}) < g(\overrightarrow{b}^0)$. *This contradicts the assumption of optimality. Thus, for the distribution that minimizes the function* $g(\overrightarrow{b}) = \max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)$, *it holds that* $\tau_2^{loc} \cdot b_2 = \tau_3^{loc} \cdot b_3 = \ldots = \tau_n^{loc} \cdot b_n$.

Let us return to the problem (3). In addition to the minimum expression already studied in (4), there are additional terms in the problem (3). δ in (3) depends on the value of b_1, but do not depend on $b_i, i = \overline{2, n}$. From this and Lemma 1, it follows that in the original problem (3), the data sharing between the 2nd, 3rd, and subsequent devices should be proportional. Thus, the problem (3) is reduced to a new problem with additional constraints:

$$\min_{\substack{\sum\limits_{i=1}^{n} b_i = N; \delta = \frac{L}{b_1{}^{\gamma}}; \\ \tau_2^{loc} \cdot b_2 = \ldots = \tau_n^{loc} \cdot b_n}} [(\max(\tau_1^{loc} \cdot b_1, \ldots, \tau_n^{loc} \cdot b_n) + \tau_{comm}) \cdot \mathcal{O}(\max\{1, \sqrt{\frac{\delta}{\mu}} \log \frac{1}{\varepsilon}\})$$
$$+ \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\max\{1, \sqrt{\frac{L}{\delta}}, \sqrt{\frac{\delta}{\mu}}, \sqrt{\frac{L}{\mu}}\} \log \frac{1}{\varepsilon})], \ \gamma \in \{\frac{1}{2}, 1\}. \tag{5}$$

3.3 Define the final minimization problem

It follows from Lemma 1 that $b_i \cdot \tau_i^{loc} = \text{const}\ \forall i \in \overline{2, n}$. Therefore,

$$N - b_1 = \sum_{i=2}^{n} b_i = \sum_{i=2}^{n} \frac{\tau_2^{loc} \cdot b_2}{\tau_i^{loc}} = \tau_2^{loc} \cdot b_2 \cdot \sum_{i=2}^{n} \frac{1}{\tau_i^{loc}} \Rightarrow b_2 = \frac{N - b_1}{\tau_2^{loc}} (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}.$$

As mentioned above, we consider the case of $\delta = \frac{L}{b_1}$ and case of $\delta = \frac{L}{\sqrt{b_1}}$.

3.3.1 Case of $\delta = \frac{L}{b_1}$. There the following relations are fulfilled:

$$\gamma = 1, \quad \mu \leq \delta \leq L \Rightarrow \begin{cases} 2 \cdot K = \mathcal{O}(\sqrt{\frac{L}{\mu b_1}} \log \frac{1}{\varepsilon}) \\ k_{some} = \mathcal{O}(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon}) \end{cases} .$$

Substituting this estimates into (5), the problem takes the following form:

$$\min_{\sum\limits_{i=1}^{n} b_i = N} [(\max\{\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2\} + \tau_{comm}) \cdot \mathcal{O}(\sqrt{\frac{L}{\mu b_1}} \log \frac{1}{\varepsilon}) + \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon})].$$

As a result, leaving the only variable b_1 in the function we pass to the final form of minimization problem:

$$\begin{aligned} \min_{0 < b_1 \leq N} \mathcal{F}(b_1) =& (\max\{\tau_1^{loc} \cdot b_1;\ (N - b_1) \cdot (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} + \tau_{comm}) \cdot \\ & \cdot \mathcal{O}(\sqrt{\frac{L}{\mu b_1}} \log \frac{1}{\varepsilon}) + \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon})]. \end{aligned} \tag{6}$$

Let us investigate the problem further. To do this, find the point at which the expressions under the maximum coincide:

$$b_1^0 \cdot (\tau_1^{loc} + (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}) = N(\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} \Rightarrow b_1^0 = \frac{N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}}{\tau_1^{loc} + (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}}.$$

Thus, we obtained two half-intervals, on each of which we can formulate a different minimization problem:

$$\begin{cases} (a): \ 0 < b_1 \leq b_1^0 \\ (b): \ b_1^0 < b_1 \leq N \end{cases}$$

$$\begin{cases} (a): \ \max\{\tau_1^{loc} \cdot b_1;\ (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} = (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} \\ (b): \ \max\{\tau_1^{loc} \cdot b_1;\ (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} = \tau_1^{loc} \cdot b_1 \end{cases}$$

We construct functions of one variable $\mathcal{F}_1(b_1), \mathcal{F}_2(b_1)$ on the corresponding half-intervals that need to be minimized according to the problem (6):

$$\begin{cases} (a): \ \mathcal{F}_1(b_1) = [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot c_1\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})b_1^{-\frac{1}{2}} - \\ \qquad - c_1\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}b_1^{\frac{1}{2}} + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})b_1 \\ (b): \ \mathcal{F}_2(b_1) = \tau_{comm} \cdot c_1\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})b_1^{-\frac{1}{2}} + c_1\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})\tau_1^{loc}b_1^{\frac{1}{2}} + \\ \qquad + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})b_1 \end{cases}.$$

Besides we can immediately find their derivatives for further analysis:

$$\begin{cases} (a): \ \mathcal{F'}_1(b_1) = -\frac{1}{2}c_1b_1^{-\frac{3}{2}}[N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon} - \\ \qquad - \frac{1}{2}c_1b_1^{-\frac{1}{2}}\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon}) \\ (b): \ \mathcal{F'}_2(b_1) = -\frac{1}{2}c_1b_1^{-\frac{3}{2}}\tau_{comm} \cdot \sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon}) + \frac{1}{2}c_1b_1^{-\frac{1}{2}}\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})\tau_1^{loc} + \\ \qquad + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon} \end{cases}.$$

3.3.2 Case of $\boldsymbol{\delta = \frac{L}{\sqrt{b_1}}}$. Here we can proceed similarly to the previous point. First, let us present the necessary relations in this case.

$$\gamma = \frac{1}{2}, \quad \mu \leq \delta \leq L \Rightarrow \begin{cases} 2 \cdot K = \mathcal{O}(\sqrt{\frac{L}{\mu\sqrt{b_1}}}\log\frac{1}{\varepsilon}) \\ k_{some} = \mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}) \end{cases}.$$

Substituting these relations into (5), we obtain

$$\min_{\sum\limits_{i=1}^{n} b_i = N} [(\max\{\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2\} + \tau_{comm}) \cdot \mathcal{O}(\sqrt{\frac{L}{\mu\sqrt{b_1}}}\log\frac{1}{\varepsilon}) + \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon})].$$

Again, getting rid of all variables except b_1 we write the final minimization problem in this case

$$\min_{0<b_1\leq N}[(\max\{\tau_1^{loc} \cdot b_1;\ (N - b_1) \cdot (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} + \tau_{comm}) \cdot \mathcal{O}(\sqrt{\frac{L}{\mu\sqrt{b_1}}}\log\frac{1}{\varepsilon}) \\ + \ \tau_1^{loc} \cdot b_1 \cdot \mathcal{O}(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon})]. \tag{7}$$

Similarly, we select the point b_1^0, it turns out to be the same as in the previous paragraph. After we can obtained two half-intervals, on each of which we can formulate a different minimization problem:

$$\begin{cases} (a): \ 0 < b_1 \leq b_1^0 \Rightarrow \max\{\tau_1^{loc} \cdot b_1; \ (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} = \\ \quad = \ (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} \\ (b): \ b_1^0 < b_1 \leq N \Rightarrow \max\{\tau_1^{loc} \cdot b_1; \ (N - b_1) \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}\} = \tau_1^{loc} \cdot b_1 \end{cases} .$$

We construct functions of one variable $\mathcal{F}_1(b_1), \mathcal{F}_2(b_1)$ on the corresponding half-intervals that need to be minimized according to problem (7):

$$\begin{cases} (a): \ \mathcal{F}_1(b_1) = [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) b_1^{-\frac{1}{4}} - \\ \quad - c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} b_1^{\frac{3}{4}} + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \cdot b_1 \\ (b): \ \mathcal{F}_2(b_1) = \tau_{comm} \cdot c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) b_1^{-\frac{1}{4}} + c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \tau_1^{loc} b_1^{\frac{3}{4}} + \\ \quad + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \cdot b_1 \end{cases} .$$

Besides immediately find their derivatives for further analysis:

$$\begin{cases} (a): \ \mathcal{F'}_1(b_1) = -\frac{1}{4} c_1 b_1^{-\frac{5}{4}} [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon} - \\ \quad - \frac{3}{4} c_1 b_1^{-\frac{1}{4}} \sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon} \\ (b): \ \mathcal{F'}_2(b_1) = -\frac{1}{4} c_1 b_1^{-\frac{5}{4}} \tau_{comm} \cdot \sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon} + \frac{3}{4} c_1 b_1^{-\frac{1}{4}} \sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \tau_1^{loc} + \\ \quad + \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon} \end{cases} .$$

3.4 Final solution

3.4.1 Case of $\boldsymbol{\delta = \frac{L}{b_1}}$. Our goal is to find the minimum of the already obtained functions $\mathcal{F}_1(b_1), \mathcal{F}_2(b_1)$. To do this, we will look for the zeros of $\mathcal{F'}_1(b_1), \mathcal{F'}_2(b_1)$. Here we obtain the cubic equation. To solve it, we can use the Cardano's formula. Consider the equation $ax^{-\frac{1}{2}} + bx^{-\frac{3}{2}} + c = 0$, where in cases $(a): \ 0 < b_1 \leq b_1^0$ and $(b): \ b_1^0 < b_1 \leq N$ we put:

$$\begin{cases} (a): \quad a = \frac{1}{2} c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}; \\ \qquad\quad b = -\frac{1}{2} c_1 [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon}; \ c = \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \\ (b): \quad a = \frac{1}{2} c_1\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \tau_1^{loc}; \\ \qquad\quad b = -\frac{1}{2} c_1 \tau_{comm} \cdot \sqrt{\frac{L}{\mu}} \log\frac{1}{\varepsilon}; \ c = \tau_1^{loc} \cdot c_2\sqrt{\frac{L}{\mu}} \log(\frac{1}{\varepsilon}) \end{cases}$$

Then on the condition that

$$\begin{aligned} N \geq & \frac{a^2}{3c^2} + \frac{\sqrt[3]{2a^6 + 3\sqrt{3}\sqrt{4a^3b^3c^6 + 27b^4c^8} + 18a^3bc^2 + 27b^2c^4}}{3\sqrt[3]{2}c^2} \\ & - \frac{\sqrt[3]{2}\left(-a^4 - 6abc^2\right)}{3c^2\sqrt[3]{2a^6 + 3\sqrt{3}\sqrt{4a^3b^3c^6 + 27b^4c^8} + 18a^3bc^2 + 27b^2c^4}}, \end{aligned}$$

we get a solution:

$$\begin{aligned} x = & \frac{a^2}{3c^2} + \frac{\sqrt[3]{2a^6 + 3\sqrt{3}\sqrt{4a^3b^3c^6 + 27b^4c^8} + 18a^3bc^2 + 27b^2c^4}}{3\sqrt[3]{2}c^2} \\ & - \frac{\sqrt[3]{2}\left(-a^4 - 6abc^2\right)}{3c^2\sqrt[3]{2a^6 + 3\sqrt{3}\sqrt{4a^3b^3c^6 + 27b^4c^8} + 18a^3bc^2 + 27b^2c^4}}. \end{aligned}$$

Hence the desired solution is trivially obtained. Since we have obtained one value of b_1 on each of the half-intervals, which is the minimum of the function on its, so by choosing the one on which the function is smaller, we obtain the optimal value of b_1.

3.4.2 Case of $\delta = \frac{L}{\sqrt{b_1}}$. Proceed similarly as in the previous paragraph does not work, since we cannot write out the solution of these equations in analytic form due to their powers. Therefore, let us consider the following particular cases:

1. $\forall i \hookrightarrow \tau_{comm} \ll \tau_i^{loc}$;
2. $\forall i \hookrightarrow \tau_{comm} \gg \tau_i^{loc}, \forall i \neq j \hookrightarrow \tau_i^{loc} = \tau_j^{loc}$.

Let us introduce new notation: $\alpha = c_1 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}, \beta = c_2 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}$, for simplicity. Now we are ready to consider two cases separately.

Case 1:

(a): $0 < b_1 \leq b_1^0$ and $\mathcal{F}_1(b_1) = [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \alpha b_1^{-\frac{1}{4}}$ $-$

$- \alpha(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} b_1^{\frac{3}{4}} + \tau_1^{loc} \cdot \beta b_1$. Let us assume that $\tau_1^{loc} \leq \tau_2^{loc} \leq \ldots \leq \tau_n^{loc}$. Using this assumption and $\tau_{comm} \ll \tau_i^{loc}$, one can obtain the following estimate:

$$\begin{aligned} (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} &= \frac{1}{\frac{1}{\tau_1^{loc}} + \ldots + \frac{1}{\tau_n^{loc}}} \\ &= \frac{\tau_2^{loc} \cdot \ldots \cdot \tau_n^{loc}}{\tau_3^{loc} \cdot \ldots \cdot \tau_n^{loc} + \tau_2^{loc} \cdot \tau_4^{loc} \cdot \ldots \cdot \tau_n^{loc} + \ldots + \tau_2^{loc} \cdot \ldots \cdot \tau_{n-1}^{loc}} \end{aligned}$$

$$\geq \frac{\tau_2^{loc}}{n-1} \gg \tau_{comm}. \tag{8}$$

Given the estimate (8), the functions $\mathcal{F}_1(b_1)$ and accordingly $\mathcal{F}'{}_1(b_1)'$ can be approximately simplified as follows:

$$\mathcal{F}_1(b_1) = \alpha(\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} \cdot b_1^{-\frac{1}{4}}(N - b_1) + \tau_1^{loc}\beta \cdot b_1,$$
$$\mathcal{F}'{}_1(b_1) = \alpha(\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} \cdot (-\frac{1}{4}b_1^{-\frac{5}{4}}N - \frac{3}{4}b_1^{-\frac{1}{4}}) + \tau_1^{loc}\beta.$$

We get the equation in the same powers, and then again we cannot write out an analytic solution, but for this problem it is easier to find a numerical solution.

(b): $b_1^0 \leq b_1 \leq N$ and $\mathcal{F}_2(b_1) = \tau_{comm} \cdot \alpha b_1^{-\frac{1}{4}} + \alpha\tau_1^{loc}b_1^{\frac{3}{4}} + \tau_1^{loc} \cdot \beta b_1 = = \alpha \cdot b_1^{-\frac{1}{4}}(\tau_{comm} + \tau_1^{loc}b_1) + \beta \cdot \tau_1^{loc} \cdot b_1$. With the same assumption that $\tau_1^{loc} \leq \tau_2^{loc} \leq \ldots \leq \tau_n^{loc}$, we get

$$\tau_1^{loc}b_1 \geq \frac{\tau_1^{loc}N\frac{\tau_2^{loc}}{n-1}}{\tau_1^{loc} + \frac{\tau_n^{loc}}{n-1}} \geq \frac{\tau_1^{loc}\tau_2^{loc}N}{(n-1)(\tau_1^{loc} + \tau_n^{loc})} \geq \frac{\tau_1^{loc}\tau_2^{loc}N}{2(n-1)\tau_n^{loc}} \gg$$
$$\gg \tau_{comm}\frac{N}{2(n-1)} \gg \tau_{comm} \tag{9}$$

Here, using (9), we can also simplify $F_2(b_1)$ and then $\mathcal{F}'{}_2(b_1)$:

$$\mathcal{F}_2(b_1) = \alpha \cdot \tau_1^{loc} \cdot b_1^{\frac{3}{4}} + \beta\tau_1^{loc} \cdot b_1, \quad \mathcal{F}'{}_2(b_1) = \frac{3}{4}\alpha \cdot \tau_1^{loc} \cdot b_1^{-\frac{1}{4}} + \beta \cdot \tau_1^{loc} > 0.$$

Since the derivative of the function is positive, the function is increasing, and therefore the minimum is taken at $b_1 = b_1^0 = \frac{N(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}}{\tau_1^{loc} + (\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}}$. Thus, in the case of small τ_{comm} we obtained the following result:

$$b_{1,\min} \leq b_1^0 = \frac{N(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}}{\tau_1^{loc} + (\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}}. \tag{10}$$

Case 2:

Here we also define: $\tau := \tau_i^{loc}\ \forall i \in 1, \ldots, n$. Then we can rewrite the target function of (6) in the following way:

$$\mathcal{F}(b_1) = (\max\{\tau b_1; (N - b_1)\frac{\tau}{n-1}\} + \tau_{comm}) \cdot \frac{\alpha}{\sqrt[4]{b_1}} + \tau\beta b_1.$$

Consider the case $\tau_{comm} = N^2\tau$. We can assume also that the size of data N is large, therefore $\tau_{comm} \gg N\tau$. And then:

$$\max\{\tau b_1; (N-b_1)\frac{\tau}{n-1}\} < \tau N \ll \tau_{comm} \Rightarrow \mathcal{F}(b_1) \approx \frac{\alpha\tau_{comm}}{\sqrt[4]{b_1}} + \beta\tau b_1$$

$$\mathcal{F}'(b_1) = -\frac{\alpha\tau_{comm}}{4b_1\sqrt[4]{b_1}} + \beta\tau = 0 \Rightarrow b_{1,\min}^{\frac{5}{4}} = \frac{\tau_{comm}\alpha}{4\beta\tau} \Rightarrow b_{1,\min} = (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}}.$$

If the found value $b_{1,\min}$ lies in the interval $(0, N)$, one can found the optimal value of $\mathcal{F}$:

$$\begin{aligned}\mathcal{F}(b_{1_{\min}}) = (\alpha\tau_{comm})^{\frac{4}{5}} \cdot (4\beta\tau)^{\frac{1}{5}} + (\beta\tau)^{\frac{1}{5}}(\frac{\alpha\tau_{comm}}{4})^{\frac{4}{5}} &= \\ = (\alpha\tau_{comm})^{\frac{4}{5}}(\beta\tau)^{\frac{1}{5}}(4^{\frac{1}{5}} + 4^{-\frac{4}{5}}).&\end{aligned}$$

Otherwise, the minimum is reached at the right boundary, since at zero we can say that the function increases. Summarizing all of the above in this case, it is worth noting that for large values of N the second special case generalizes to the following condition:

$$\forall i \hookrightarrow \tau_{comm} = \mathcal{O}(N^k\tau_i^{loc}) \text{ with } k > 1, \text{ and } \forall i \neq j \hookrightarrow \tau_i^{loc} = \tau_j^{loc} = \tau,$$
$$\min \mathcal{F}(b_1) = \begin{cases} (\alpha\tau_{comm})^{\frac{4}{5}} \cdot (\beta\tau)^{\frac{1}{5}}(4^{\frac{1}{5}} + 4^{-\frac{4}{5}}), & 0 < (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}} < N \\ \frac{\alpha\tau_{comm}}{N} + \beta\tau N, & (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}} \geq N \end{cases}. \quad (11)$$

3.5 Numerical solution

Since an analytical solution is not found for all cases, we can give a general numerical solution to our problem. In order to determine the minimum of these functions on the respective half-intervals, we examine points where the derivatives of $\mathcal{F'}_1(b_1)$ and $\mathcal{F'}_2(b_1)$ approach zero. It should be noted that, given the nature of these functions, their derivatives can only be zero once on the desired half-interval. Hence, by employing the Newton's method [14] for $\mathcal{F'}_1(b_1)$ and $\mathcal{F'}_2(b_1)$, we can locate its zeros. Subsequently, we need to compare the values of the corresponding function at these points with the value at the extreme point of the interval. One of these points provides the optimal value, thereby serving as the ultimate solution to the problem (6) and (7).

4 Noise in the networks

Now we proceed to the third assumption from Section 1.3. As mentioned above, let τ_{comm} and τ_i^{loc} be random variables with $\mathbb{E}[\tau_{comm}] < \infty$, $\mathbb{E}[\tau_i^{loc}] < \infty$, $\mathbb{D}[\tau_{comm}] < \infty$, $\mathbb{D}[\tau_i^{loc}] < \infty$. Here we consider the case of $\delta = \frac{L}{\sqrt{b_1}}$. The analytical solution in this setup was obtained in the particular cases for small and large communication times. Let us consider them separately.

4.1 Case of big communication time

Let us consider the case of (11). In particular, we obtained that

$$\mathcal{F}(b_{1,\min}) = (\alpha \cdot \tau_{comm})^{4/5} \cdot (\beta \cdot \tau_1^{loc})^{1/5} \cdot (4^{1/5} + 4^{-4/5}) \tag{12}$$

with $\alpha = c_1 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}, \beta = c_2 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}$.

To estimate the variance of the result $\mathcal{F}(b_{1_{\min}})$, we produce the following equality: $X, Y-$ independent random variables $\Rightarrow \mathbb{D}[XY] = \mathbb{E}[(XY - \mathbb{E}[XY])^2] = \mathbb{E}[(XY)^2] - 2\mathbb{E}^2[XY] + \mathbb{E}^2[XY] = \mathbb{E}[X^2]\mathbb{E}[Y^2] - \mathbb{E}^2[X]\mathbb{E}^2[Y] = (\mathbb{D}[X] + \mathbb{E}^2[X]) \cdot (\mathbb{D}[Y] + \mathbb{E}^2[Y]) - \mathbb{E}^2[X]\mathbb{E}^2[Y] = \mathbb{D}[X]\mathbb{D}[Y] + \mathbb{D}[X]\mathbb{E}^2[Y] + \mathbb{D}[Y]\mathbb{E}^2[X]$. Applying this equality to (12), we get the variance of $\mathcal{F}(b_{1_{\min}})$:

$$\begin{aligned}\mathbb{D}[\mathcal{F}(b_{1,\min})] =& [\alpha^{4/5} \cdot \beta^{1/5} \cdot (4^{1/5} + 4^{-4/5})] \cdot \{\mathbb{D}[(\tau_{comm})^{4/5}]\mathbb{D}[(\tau_1^{loc})^{1/5}] \\ &+ \mathbb{D}[(\tau_{comm})^{4/5}]\mathbb{E}^2[(\tau_1^{loc})^{1/5}] + \mathbb{D}[(\tau_1^{loc})^{1/5}]\mathbb{E}^2[(\tau_{comm})^{4/5}]\}.\end{aligned}$$

4.2 Case of small communication time

Here we consider noise only in communications, i.e. we put the time of communications as a random variable with mathematical expectation and finite variance, and the time of local computations for each device as a constant value. We consider the function value (7) at the point (10):

$$\mathcal{F}(b_{1,\min}) = [(N - b_1^0) \cdot (\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \alpha \cdot \frac{1}{(b_1^0)^{1/4}} + \tau_1^{loc} \cdot b_1^0 \cdot \beta \ =$$

$$= \left[\frac{\tau_1^{loc} \cdot N \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}}{\tau_1^{loc} + (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1}} + \tau_{comm}\right] \cdot \alpha \cdot \frac{(\tau_1^{loc} + (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1})^{1/4}}{(N \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1})^{1/4}} + \tau_1^{loc} \cdot b_1^0 \cdot \beta$$

with $\alpha = c_1 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}, \beta = c_2 \cdot \sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}$.

Then the required variance is as follows:

$$\mathbb{D}[\mathcal{F}(b_{1,\min})] = \left[\alpha \cdot \frac{(\tau_1^{loc} + (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1})^{1/4}}{(N \cdot (\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1})^{1/4}}\right]^2 \cdot \mathbb{D}[\tau_{comm}].$$

5 Experiments

5.1 Experiments with data distribution

For experimental verification of the theoretical results we consider the ridge regression problem:

$$\min_{\omega}[\frac{1}{2N}\|X\omega - y\|_2^2 + \frac{\lambda}{2}\|\omega\|_2^2], \tag{13}$$

where ω is the vector of weights of the model, $\{x_i, y_i\}_{i=1}^N$ is the training dataset, and $\lambda > 0$ is the regularization parameter. We consider a network with 21 workers simulated on a single-CPU machine. We use dataset from LIBSVM library [3]. Value $\tau_1^{loc} = 1$, values for other i $\tau_i^{loc}, i \neq 1$ were taken conditionally and generated uniformly from 3 to 7. τ_{comm} were chosen so that $\frac{\tau_{comm}}{\tau_1^{loc}} = 10^l, l = \overline{-6, 12}$

Remark 1. In the first stages $\arg\min\limits_x [p(x_k^g) + < \nabla p(x_k^g), x - x_k^g > + \frac{1}{2\Theta}||x - x_k^g||^2 + q(x)]$ was searched explicitly (Line 5 of Algorithm 1). This was done by equating the gradient to zero. Applying to the problem 13:

$$\nabla p(\omega_k^g) + \frac{1}{\Theta}(x - \omega_k^g) + \nabla q(x) = 0 \Rightarrow \lambda\omega_k^g + \frac{1}{\Theta}(Ix - \omega_k^g) + \frac{1}{N}X^T(Xx - y) = 0 \Rightarrow$$
$$x = (I\frac{1}{\Theta} + \frac{1}{N}X^TX)^{-1}(\frac{1}{\Theta}\omega_k^g + \frac{1}{N}X^Ty - \lambda\omega_k^g)$$

Then in Algorithm 1 on line 5: $x_f^{k+1} = x$

We implement Algorithm 1 in Python 3.9.6 using the iterative OGM-G method from [8] to find the arg min in 1 (that is what the original article [10] recommends). After calculating the required number of iterations to achieve a certain accuracy we find the values of constants c_1, c_2, and, respectively, α, β. With their help, we are able to distribute the data from the dataset to the devices according to the above formulas.

Next, we run the algorithm and measure the running time on the resulting distribution of data across devices and uniform distribution. Our goal is find the acceleration between our choice of data distribution and uniform splitting.

Two cases of different δ: $\delta = \frac{L}{\sqrt{b_1}}$, $\delta = \frac{L}{b_1}$, are considered. For the case with $\delta = \frac{L}{\sqrt{b_1}}$ we use following approaches to find $b_{1,\min}$: 1) for all cases of the communication time we use the Newton's method to find the solution numerically; 2) for small and large communications we also use results of Section 3.4.2. For the case with $\delta = \frac{L}{b_1}$ to find $b_{1,\min}$ we also use the Newton's method and additionally the Cardano's formula from Section 3.4.1. See Figure 1 for results.

Let us analyze the obtained plots. The formula for the case of large communications (Section 3.4.2. ,Case 2) and the Cardano's formula (Section 3.4.1.) practically coincided with the optimal solution search by the Newton's method. The case of small communications showed worse results. This is explained by the fact that the formula was obtained in rough approximation.

5.2 Experiments with noise

We modify the simulation of Algorithm 1 by adding noise to communication and device power times. We generate the noise from a uniform distribution and its values are 10, 20, 30, 50, and 100 percent, respectively, relative to the absolute value of communication time and device power. In the new noise model, we measure the running time of the ridge regression problem with the resulting data distribution and with the uniform one, obtaining the acceleration that gives our data distribution. In the process, we measure the mathematical expectation of

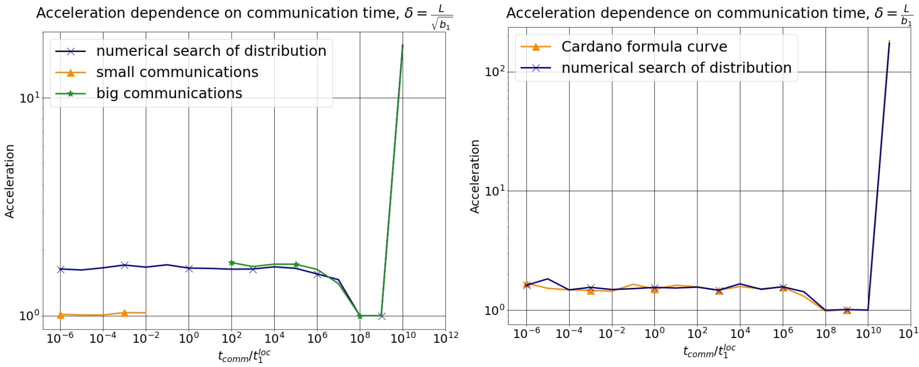

Fig. 1: Experiments with data distribution

communication costs and device powers and obtained the acceleration at these expected values. Experiments are conducted only in the case of big communication costs. In Figure 2, we plots the ratio of these accelerations and also confidence intervals.

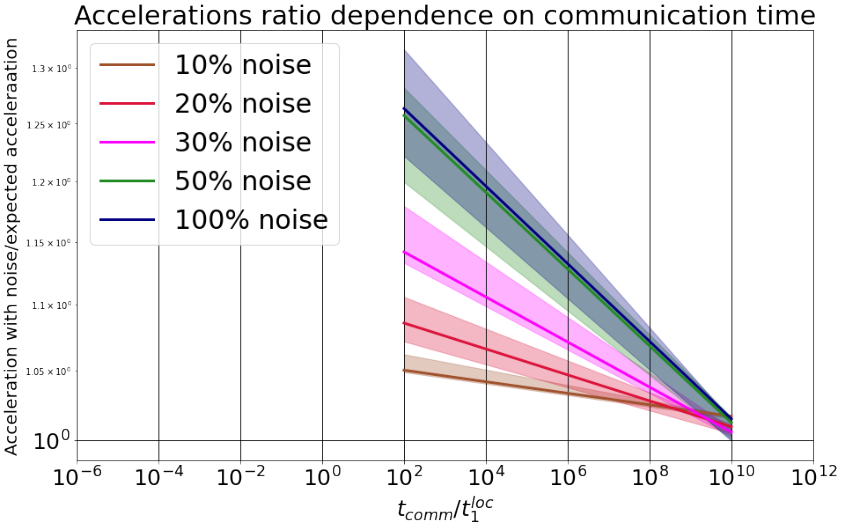

Fig. 2: Experiments with the noise in the network

Let us analyze the obtained results. The figure shows that all the lines fall within the confidence intervals, which means that for any value of noise, the theoretical calculations given in Section 4.1 are confirmed by experiments. We also note that near the value $t_{comm}/t_1^{loc} = 10^{10}$, noise practically ceases to affect the results.

6 Conclusion

In this article, we presented a new data partioning method for a distributed optimization problem with different time costs of local computations and communications.e. Our solution is based on constructing the running time function of Algorithm 1 and finding its minimum. Our method works well in networks with varying communication costs between the server and local devices and different capacities of the devices. The theoretical results confirmed experimentally. This shows that our method gives acceleration on this type of problems. In addition, by assuming noise in the networks, we obtained the error of the optimal solution and conducted appropriate experiments.

References

1. Arjevani, Y., Shamir, O.: Communication complexity of distributed convex learning and optimization. Advances in neural information processing systems **28** (2015)
2. Beznosikov, A., Scutari, G., Rogozin, A., Gasnikov, A.: Distributed saddle-point problems under data similarity. Advances in Neural Information Processing Systems **34**, 8172–8184 (2021)
3. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST) **2**(3), 1–27 (2011)
4. Ghosh, A., Maity, R.K., Mazumdar, A., Ramchandran, K.: Communication efficient distributed approximate newton method. In: 2020 IEEE International Symposium on Information Theory (ISIT). pp. 2539–2544. IEEE (2020)
5. Gorbunov, E., Burlachenko, K.P., Li, Z., Richtárik, P.: Marina: Faster non-convex distributed learning with compression. In: International Conference on Machine Learning. pp. 3788–3798. PMLR (2021)
6. Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., Massoulie, L.: Statistically preconditioned accelerated gradient method for distributed optimization. In: International conference on machine learning. pp. 4203–4227. PMLR (2020)
7. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)
8. Kim, D., Fessler, J.A.: Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. Journal of optimization theory and applications **188**(1), 192–219 (2021)
9. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
10. Kovalev, D., Beznosikov, A., Borodich, E., Gasnikov, A., Scutari, G.: Optimal gradient sliding and its application to optimal distributed optimization under similarity. Advances in Neural Information Processing Systems **35**, 33494–33507 (2022)
11. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine **37**(3), 50–60 (2020)

12. Matsushima, S., Yun, H., Zhang, X., Vishwanathan, S.: Distributed stochastic optimization of the regularized risk. arXiv preprint arXiv:1406.4363 (2014)
13. Nesterov, Y., et al.: Lectures on convex optimization, vol. 137. Springer (2018)
14. Polyak, B.T.: Newton's method and its use in optimization. European Journal of Operational Research **181**(3), 1086–1096 (2007)
15. Reddi, S.J., Konečnỳ, J., Richtárik, P., Póczós, B., Smola, A.: Aide: Fast and communication efficient distributed optimization. arXiv preprint arXiv:1608.06879 (2016)
16. Shamir, O., Srebro, N., Zhang, T.: Communication-efficient distributed optimization using an approximate newton-type method. In: International conference on machine learning. pp. 1000–1008. PMLR (2014)
17. Smith, V., Forte, S., Chenxin, M., Takáč, M., Jordan, M.I., Jaggi, M.: Cocoa: A general framework for communication-efficient distributed optimization. Journal of Machine Learning Research **18**, 230 (2018)
18. Sun, Y., Scutari, G., Daneshmand, A.: Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation. SIAM Journal on Optimization **32**(2), 354–385 (2022)
19. Tian, Y., Scutari, G., Cao, T., Gasnikov, A.: Acceleration in distributed optimization under similarity. In: International Conference on Artificial Intelligence and Statistics. pp. 5721–5756. PMLR (2022)
20. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., Rellermeyer, J.S.: A survey on distributed machine learning. Acm computing surveys (csur) **53**(2), 1–33 (2020)