Optimal data splitting in distributed optimization for machine learning

Technical Report

Gleb Molodtsov
MIPT, Russia
molodtsov.gl@phystech.edu

Daniil Medyakov
MIPT, Russia
mediakov.do@phystech.edu

Alexander Beznosikov
MIPT, Russia
anbeznosikov@gmail.com

Abstract

The distributed optimization problem has become increasingly relevant recently. We consider this problem in the context of varying capacities of devices between which data is shared. The objective of this study is to achieve an optimal ratio of distributed data between the server and local machines. Optimal gradient descent and its application to distributed optimization under similarity are employed to address this problem. However, most distributed approaches suffer from a significant bottleneck - the cost of communications. The paper proposes a solution that takes into account this cost. The running times of the system are compared between uniform and optimal distributions. The superior theoretical performance of our solutions is experimentally validated.

1 Introduction

1.1 Distributed optimization

To achieve the best results in modern machine learning and minimization tasks, researchers and practitioners face various challenges. Training modern machine learning models remains an extremely challenging task, also because models are trained on increasingly large datasets. Having more data in the dataset increases the robustness and generalizability of the trained 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 devices of the network, such as in federated learning. 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. We consider optimization problems of the following form:

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

Here, $x \in \mathbb{R}^d$ collects the parameters of a statistical model to be trained, n is the number of devices, and f_i represents the convex loss-function of agent i, which is unknown to the other agents. Several solution methods have been proposed to solve (1). The prototype approach involves interleaving edge devices calculations (nodes $i = 1, \dots n$) with communications to and from the master node ($i = 1$). The master node maintains and updates the authoritative copy of the optimization variables, eventually producing the final solution estimate.

1.2 Distributed optimization under similarity

Since communication cost often becomes the bottleneck in distributed computing, significant research has focused on developing communication-efficient distributed algorithms. Acceleration, based on Nesterov's concept, has been extensively studied for reducing the communication burden. For L-smooth and μ-strongly convex functions r in (1), first-order methods guarantee linear convergence with computation and communication complexities proportional to $\sqrt{\kappa}$. Here, $\kappa := {}^{L}\!/\!{}_{\mu}$ represents the condition number of r. However, for ill-conditioned functions with a large κ, the polynomial dependence on κ may be unsatisfactory. 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 function similarity. This problem is considered in [1], [8], [6]. 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. Consider for example a quadratic problem: $\nexists \ \delta : ||(A_i - A_j)x|| < \delta \ if \ x \to \infty$. Therefore, we will consider a different approach. Specifically, for all x in a suitable domain of interest and all $i \neq j = 1, \dots, n$, 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.

1.3 Contributions

For this problem, as shown above, there already exist algorithms such as [10], [9], [7], [3], [2]. The last word was an accelerated extragradient algorithm [5], which reached the lower estimates [1]. But all of them do not take into account in any way the fact that devices in the network may have different capacities, i.e., process a unit of information in different times. In this paper, we reply the question:

Can we find such a distribution of data among the devices in the network to reduce the actual running time of the optimal algorithm [5]*?*

Our contribution is 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 [5] and takes into account the different capacity of edge devices.
- **Comprehensive analysis:** We pay special attention to the limiting cases and obtain results in them. The case where communications are too expensive is not of practical interest as the whole idea of distributed learning is lost, but the case of inexpensive communications (not so expensive that the communication takes longer than processing all data by just one device) is of great interest.
- **Different techniques for obtaining a solution:** We obtain results in different cases, including for different estimates of δ. We use different techniques: Cardano's formula, upper estimates in limiting cases, finding the zero of the function using the simplest numerical methods.

We also conducted experiments confirming that with the obtained distribution it takes less time to solve the selected problem.

2 Problem Statement

To achieve lower communication and local gradient complexity, we can refer to Algorithm 1 from [5]. For this purpose, the function r needs to be transformed into the following form:

$$r(x) = \underbrace{f_1(x)}_{q(x)} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}[f_i(x) - f_1(x)],}_{p(x)} \tag{2}$$

Here, r is assumed to be convex and decomposed as the sum of a smooth, potentially non-convex function p and a smooth convex function q. First-order information of p and q can be accessed separately. We are particularly interested in scenarios where evaluating the gradients of these two functions incurs heterogeneous costs, i.e., the cost of computing ∇p is significantly higher than that of computing ∇q.

Here is this algorithm .

Algorithm 1 Accelerated Extragradient

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

Let us calculate how many operations this algorithm performs per iteration. In line 5, there is one communication, one local computation, one central node calculation, and additional central node computations. In line 6, there is one communication, one local computation, and one central node calculation. We make the following notations: τ_i - the time of one local computation on i-th device, K - the number of iterations, τ_{comm} - the time for one communication, k_{some} - additional computations of the central node, n - the number of nodes in the network. Then we can write the general running time of the algorithm as:

$$T_{sum} = 2 \cdot \max(\tau_1, \tau_2, \dots, \tau_n) \cdot K + 2 \cdot K \cdot \tau_{comm} + \tau_1 \cdot k_{some} \tag{3}$$

Our task is to minimize the time T_{sum}. Let's 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. b_i must 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 $\delta = \frac{L}{b_i}$ (this estimate is given in [5]). We obtained the following optimization problem:

$$\min_{\sum\limits_{i=1}^{n} b_i = N; \delta = \frac{L}{b_1{}^{\gamma}}} [2 \cdot \max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n) \cdot K + 2 \cdot K \cdot \tau_{comm} + \tau_1 \cdot k_{some}], \; \gamma \in \{\frac{1}{2}, 1\} \tag{4}$$

3 How to solve (4)

3.1 The primary problem of minimization

In [5] the estimates of K and k_{some} are found, namely:
$2 \cdot K = \mathcal{O}(\max\{1, \sqrt{\frac{L_p}{\mu}}\} log(\frac{1}{\varepsilon})), \tau_1 \cdot k_{some} = \mathcal{O}(\max\{1, \sqrt{\frac{L_q}{L_p}}, \sqrt{\frac{L_p}{\mu}}, \sqrt{\frac{L_q}{\mu}}\} \log(\frac{1}{\varepsilon}))$.
The value of τ_{comm} will be determined later.

Thus, (4) is reduced to:

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

$$+ \mathcal{O}(\max\{1, \sqrt{\frac{L_q}{L_p}}, \sqrt{\frac{L_p}{\mu}}, \sqrt{\frac{L_q}{\mu}}\} \log(\frac{1}{\varepsilon}))],\ \gamma \in \{\frac{1}{2}, 1\}$$

3.2 Auxiliary problem

Consider an auxiliary problem:

$$\min_{\sum\limits_{i=2}^{n} b_i = N} [\max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)] \tag{6}$$

Lemma 1. *The solution of problem* (6) *is* $\overrightarrow{b}$ *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 $g(b) = \max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)$.
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
$b_i^0 > N(1 + \tau_i^{loc} \sum\limits_{\substack{j=2 \\ j \neq i}}^{n} \frac{1}{\tau_j^{loc}})^{-1}$.

Next, let us consider $b_i = N(1 + \tau_i^{loc} \sum\limits_{\substack{j=2 \\ j \neq i}}^{n} \frac{1}{\tau_j^{loc}})^{-1}, \quad b_j = \frac{\tau_i^{loc}}{\tau_j^{loc}} \cdot b_i\ \forall j \in \{2, \ldots, n\}$. 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 minimality.
Thus, for the distribution that minimizes the function $g(b) = \max(\tau_2^{loc} \cdot b_2, \tau_3^{loc} \cdot b_3, \ldots, \tau_n^{loc} \cdot b_n)$, 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 (5). In addition to the minimum expression already studied in the problem (6), there are additional terms in the problem (5). Note that $L_q = L, L_p = \delta = \frac{L}{\sqrt{b_1}}$ or

$L_p = \delta = \frac{L}{b_1}$(this estimate is given in [5]). They depend on the value of b_1, but do not depend on $b_i, i \in \overline{2,n}$. From this and Lemma 1, it follows that in the original problem (5), the data sharing between the 2nd, 3rd, and subsequent devices should be proportional.Thus, the problem (5) is reduced to a new problem with additional constraints:

$$\min_{\substack{\sum\limits_{i=1}^{n} b_i=N;\delta=\frac{L}{b_1{}^{\gamma}};\\ \tau_2^{loc}\cdot b_2=\ldots=\tau_n^{loc}\cdot b_n}} [(\max(\tau_1^{loc}\cdot b_1, \tau_2^{loc}\cdot b_2, \ldots, \tau_n^{loc}\cdot b_n) + \tau_{comm})\cdot \mathcal{O}(\max\{1, \sqrt{\frac{L_p}{\mu}}\log(\frac{1}{\varepsilon})\} \quad (7)$$

$$+ \mathcal{O}(\max\{1, \sqrt{\frac{L_q}{L_p}}, \sqrt{\frac{L_p}{\mu}}, \sqrt{\frac{L_q}{\mu}}\}\log(\frac{1}{\varepsilon}))],\ \gamma \in \{\frac{1}{2}, 1\}$$

3.3 Define the final minimization problem

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

$$L_p = \delta, L_q = L, \mu \le \delta \le L \Rightarrow \begin{cases} 2\cdot K = \mathcal{O}(\sqrt{\frac{L_p}{\mu}}\log(\frac{1}{\varepsilon})) = \mathcal{O}(\sqrt{\frac{L}{\mu b_1}}\log(\frac{1}{\varepsilon})) \\ \tau_1\cdot k_{some} = \mathcal{O}(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon})) \end{cases}.$$

Substituting the estimates into (7) the problem will take 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:

$$\min_{0<b_1\le N}[(\max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} + \tau_{comm})\cdot\mathcal{O}(\sqrt{\frac{L}{\mu b_1}}\log(\frac{1}{\varepsilon})) \quad (8)$$

$$+ \tau_1^{loc}\cdot b_1\cdot\mathcal{O}(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon}))]$$

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:

$$\begin{cases}(a)\ \ 0 < b_1 \leq b_1^0 \Rightarrow \max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} = (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\\ (b)\ \ b_1^0 < b_1 \leq N \Rightarrow \max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} = \tau_1^{loc}\cdot b_1\end{cases}. \quad (9)$$

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 (8). Besides immediately find their derivatives for further analysis.

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

$(b):\ \mathcal{F}_2(b_1) = \tau_{comm}\cdot c_1\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})b_1^{-\frac{1}{2}} + c_1\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})\tau_1^{loc}b_1^{\frac{1}{2}} + \tau_1^{loc}\cdot c_2\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})b_1$

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

$(b):\ \mathcal{F'}_2(b_1) = -\frac{1}{2}c_1b_1^{-\frac{3}{2}}\tau_{comm}\cdot\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon}) + \frac{1}{2}c_1b_1^{-\frac{1}{2}}\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})\tau_1^{loc} + \tau_1^{loc}\cdot c_2\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})$

3.3.2 Case of $\delta = \frac{L}{\sqrt{b_1}}$

We will proceed similarly to the previous point. First, let us present the necessary relations in this case.

$$L_p = \delta, L_q = L, \mu \leq \delta \leq L \Rightarrow \begin{cases}2\cdot K = \mathcal{O}(\sqrt{\frac{L_p}{\mu}}\log(\frac{1}{\varepsilon})) = \mathcal{O}(\sqrt{\frac{L}{\mu\sqrt{b_1}}}\log(\frac{1}{\varepsilon}))\\ \tau_1\cdot k_{some} = \mathcal{O}(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon}))\end{cases}.$$

Substituting these relations into (7) 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

$$\min_{0<b_1\leq N}[(\max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} + \tau_{comm})\cdot\mathcal{O}(\sqrt{\frac{L}{\mu\sqrt{b_1}}}\log(\frac{1}{\varepsilon})) \quad (10)$$

$$+\ \tau_1^{loc}\cdot b_1\cdot\mathcal{O}(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\varepsilon}))]$$

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:

$$\begin{cases}(a)\ \ 0 < b_1 \leq b_1^0 \Rightarrow \max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} = (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\\ (b)\ \ b_1^0 < b_1 \leq N \Rightarrow \max\{\tau_1^{loc}\cdot b_1;\ (N-b_1)\cdot(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}\} = \tau_1^{loc}\cdot b_1\end{cases} \quad (11)$$

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 (10). Besides immediately find their derivatives for further analysis.

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

$(b)\ :\ \mathcal{F}_2(b_1) = \tau_{comm}\cdot c_1\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})b_1^{-\frac{1}{4}} + c_1\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})\tau_1^{loc}b_1^{\frac{3}{4}} + \tau_1^{loc}\cdot c_2\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})b_1$

$(a)\ :\ \mathcal{F'}_1(b_1) = -\frac{1}{4}c_1b_1^{-\frac{5}{4}}[N(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1}+\tau_{comm}]\cdot\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon}) - \frac{3}{4}c_1b_1^{-\frac{1}{4}}\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})(\sum\limits_{i=2}^{n}\frac{1}{\tau_i^{loc}})^{-1} + \tau_1^{loc}\cdot c_2\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})$

$(b)\ :\ \mathcal{F'}_2(b_1) = -\frac{1}{4}c_1b_1^{-\frac{5}{4}}\tau_{comm}\cdot\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon}) + \frac{3}{4}c_1b_1^{-\frac{1}{4}}\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})\tau_1^{loc} + \tau_1^{loc}\cdot c_2\sqrt{\frac{L}{\mu}}log(\frac{1}{\varepsilon})$

3.4 Final solution in limiting cases

3.4.1 Case of $\delta = \frac{L}{b_1}$

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 formula.
Consider the equation $ax^{-\frac{1}{2}} + bx^{-\frac{3}{2}} + c = 0$,
where in cases $(a):\ \ 0 < b_1 \le b_1^0$ and $(b):\ \ b_1^0 < b_1 \le N$ we assume:

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

Then on the condition that

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

We get a solution:

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

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 desired b_1.

3.4.2 Case of $\delta = \frac{L}{\sqrt{b_1}}$

Proceed similarly to 3.4.1 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 limiting cases:

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

Establish $\alpha = c_1 \cdot \sqrt{\frac{L}{\mu}} \cdot \log(\frac{1}{\varepsilon}), \beta = c_2 \cdot \sqrt{\frac{L}{\mu}} \cdot \log(\frac{1}{\varepsilon})$

Consider case 1:

a) $0 < b_1 \leq b_1^0$
$\mathcal{F}_1(b_1) = [N(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} + \tau_{comm}] \cdot \alpha b_1^{-\frac{1}{4}} - \alpha(\sum\limits_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} b_1^{\frac{3}{4}} + \tau_1^{loc} \cdot \beta b_1$
Consider

$$\tau_1^{loc} \leq \tau_2^{loc} \leq \ldots \leq \tau_n^{loc} \tag{12}$$

Make the following estimate:

$$(\sum_{i=2}^{n} \frac{1}{\tau_i^{loc}})^{-1} = \frac{1}{\frac{1}{\tau_1^{loc}} + \ldots + \frac{1}{\tau_n^{loc}}} = \tag{13}$$

$$= \frac{\tau_2^{loc} \cdot \ldots \cdot \tau_n^{loc}}{\tau_3^{loc} \cdot \ldots \cdot \tau_n^{loc} + \tau_2^{loc} \cdot \tau_4^{loc} \cdot \ldots \cdot \tau_n^{loc} + \ldots + \tau_2^{loc} \cdot \ldots \cdot \tau_{n-1}^{loc}} \underset{(12)}{\geq} \frac{\tau_2^{loc}}{n-1} \gg \tau_{comm}$$

Given the estimate (13), the functions $\mathcal{F}_1(b_1), \mathcal{F'}_1(b_1)'$ are transformed 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 so again we cannot write out an analytic solution.

b) $b_1^0 \leq b_1 \leq N$
$\mathcal{F}_2(b_1) = \tau_{comm} \cdot \alpha b_1^{-\frac{1}{4}} + \alpha \tau_1^{loc} b_1^{\frac{3}{4}} + \tau_1^{loc} \cdot \beta b_1 = \alpha \cdot b_1^{-\frac{1}{4}}(\tau_{comm} + \tau_1^{loc} b_1) + \beta \cdot \tau_1^{loc} \cdot b_1$
Make the following estimate

$$\tau_1^{loc} b_1 \underset{b_1 \geq b_1^0, (12)}{\geq} \frac{\tau_1^{loc} N \frac{\tau_2^{loc}}{n-1}}{\tau_1^{loc} + \frac{\tau_n^{loc}}{n-1}} \geq \frac{\tau_1^{loc} \tau_2^{loc} N}{(n-1)(\tau_1^{loc} + \tau_n^{loc})} \geq \frac{\tau_1^{loc} \tau_2^{loc} N}{2(n-1)\tau_n^{loc}} \gg \tag{14}$$

$$\gg \tau_{comm} \frac{N}{2(n-1)} \gg \tau_{comm}$$

Then taking into account (14):

$$\mathcal{F}_2(b_1) = \alpha \cdot \tau_1^{loc} \cdot b_1^{\frac{3}{4}} + \beta \tau_1^{loc} \cdot b_1$$

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

Since the derivative of the function is positive, the function is increasing, and therefore the minimum will be 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:

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

Consider case 2:

Establish: $\tau := t_i^{loc} \; \forall i \in 1, \ldots, n$.

$$\text{Then } \mathcal{F} = (\max\{\tau b_1; (N - b_1)\frac{\tau}{n-1}\} + \tau_{comm}) \cdot \frac{\alpha}{\sqrt[4]{b_1}} + \tau\beta b_1 \qquad (15)$$

Consider the case $\tau_{comm} = N^2\tau$. N can be considered large, so $\tau_{comm} \gg N\tau$. Then:

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

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

Assuming that the found value b_1 lies on the interval $(0, N)$, that is, at $0 < (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}} < N$, it will be the point of minimum function $\mathcal{F}$. Then:

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

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

$$\forall i \hookrightarrow \tau_{comm} = \mathcal{O}(N^k\tau_i^{loc}), k > 1, \forall i \neq j \hookrightarrow \tau_i^{loc} = \tau_j^{loc} \qquad (16)$$

$$\min\ \mathcal{F}(b_1) = \begin{cases} (\alpha\tau_{comm})^{\frac{4}{5}} \cdot (4\beta\tau)^{\frac{1}{5}}(4^{\frac{1}{5}} + 4^{-\frac{4}{5}}), 0 < (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}} < N \\ \frac{\alpha\tau_{comm}}{N} + \beta\tau N, (\frac{\tau_{comm}\alpha}{4\beta\tau})^{\frac{4}{5}} \geq N \end{cases} . \qquad (17)$$

3.5 Practical solution

Since an analytical solution is not found for all cases, we give a general numerical solution to our problem. In order to determine the minimum of these functions on the respective half-intervals, we will 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 basic methods, we can locate the zeros of the derivatives. 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 will provide the minimum solution, thereby serving as the ultimate solution to the problem (8) and (10).

4 Experiments

4.1 Description of experiments

For experimental verification of the theoretical results we consider the problem "Ridge Regression":

$$\min_{\omega}[\frac{1}{2N}X\omega - y^2 + \frac{\lambda}{2}\omega^2], q(\omega) = \frac{1}{2N}X\omega - y^2, p(\omega) = \frac{\lambda}{2}\omega^2 \qquad (18)$$

The file a9a.txt with number of lines N = 97683 was chosen as dataset.

We implemented Algorithm 1 in the Python. The iterative OGM-G method of [4] was applied to find the solution of line 5 of Algorithm 1. 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 were able to distribute the data from the dataset to the devices according to the above formulas.

Next, we ran the algorithm and measured the running time on the resulting distribution of data across devices and uniform distribution. The cases of large and small communications were considered.

In the end, two problems were considered:

1. $\delta = \frac{L}{\sqrt{b_1}}$
2. $\delta = \frac{L}{b_1}$

For problem 1, following cases were considered:

1. small communications ($b_1 = b_1^0$) (3.4.2)
2. large communications (3.4.2)
3. search for optimal allocation using Python optimization tools (3.5)

For problem 2, following cases were considered:

1. search for the optimal solution using the Cardano formula (3.4.1)
2. search for optimal allocation using Python optimization tools (3.5)

For all cases, acceleration was found and graphs were plotted at 1

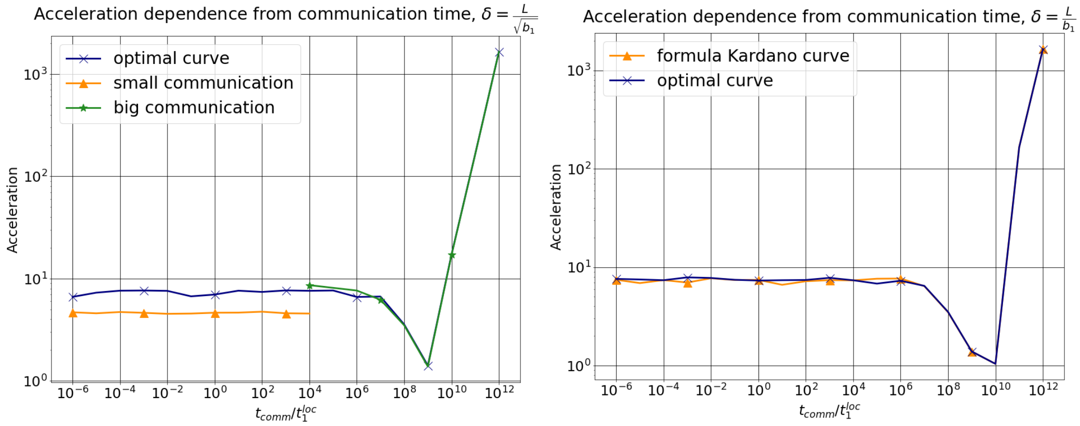

Figure 1: Final results

4.2 Analyze

Let us analyze the obtained graphs. The formula for the case of large communications and the Cardano formula practically coincided with the optimal solution search. The case of small communications showed worse results. This is explained by the fact that the formula was obtained in rough approximation. But if we take into account the constants α, β, we can get a better result, which is shown below.

$$F = \left(\max\left\{\tau_1^{loc}\cdot b_1; (N-b_1)\cdot\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\right\} + \tau_{comm}\right)\cdot\frac{\alpha}{\sqrt[4]{b_1}} + \tau_1^{loc}b_1\cdot\beta,$$

It has already been evaluated that $b_1 \leq b_1^0 \Rightarrow F = (N-b_1)\cdot\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\cdot\frac{\alpha}{\sqrt[4]{b_1}} + \tau_1^{loc}b_1\cdot\beta$

$$F = N\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\alpha b^{-\frac{1}{4}} - \left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\alpha b_1^{\frac{3}{4}} + \tau_1^{loc}\beta b_1$$

$$\text{Consider that}\quad \alpha \sim 10^6, \beta \sim 10^9 \Rightarrow$$

$$F \cong 10^6 N\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\cdot b^{-\frac{1}{4}} - 10^6\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\cdot b_1^{\frac{3}{4}} + 10^9\tau_1^{loc} b_1$$

$$\frac{1}{4}10^6 N\left(\sum_{i=2}^{n}\frac{1}{\tau_i^{loc}}\right)^{-1}\cdot b_1^{-\frac{5}{4}} \le 10^5\tau_1^{loc} \Rightarrow b_1 \le \frac{4\cdot 10^3\tau_1^{loc}}{N\left(\sum_{i=2}^{n}\left(\tau_i^{loc}\right)^{-1}\right)^{-1}}$$

5 Conclusion

In this paper we presented a new way to partition the data for the distributed optimization problem. Our solution is based on separating convex and non-convex functions and applying Algorithm 1 as well as the OGM-G algorithm from [4]. Our method works well on networks with various communication costs between the server and the local devices. The theoretical results have been confirmed experimentally. This indicates that our method gives acceleration on tasks of this type.

References

[1] Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *Advances in neural information processing systems*, 28, 2015.

[2] Aleksandr Beznosikov, Gesualdo Scutari, Alexander Rogozin, and Alexander Gasnikov. Distributed saddle-point problems under data similarity. *Advances in Neural Information Processing Systems*, 34:8172–8184, 2021.

[3] Hadrien Hendrikx, Lin Xiao, Sebastien Bubeck, Francis Bach, and Laurent Massoulie. Statistically preconditioned accelerated gradient method for distributed optimization. In *International conference on machine learning*, pages 4203–4227. PMLR, 2020.

[4] Donghwan Kim and Jeffrey A Fessler. 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.

[5] Dmitry Kovalev, Aleksandr Beznosikov, Ekaterina Borodich, Alexander Gasnikov, and Gesualdo Scutari. Optimal gradient sliding and its application to optimal distributed optimization under similarity. *Advances in Neural Information Processing Systems*, 35:33494–33507, 2022.

[6] Shin Matsushima, Hyokun Yun, Xinhua Zhang, and SVN Vishwanathan. Distributed stochastic optimization of the regularized risk. *arXiv preprint arXiv:1406.4363*, 2014.

[7] Sashank J Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

[8] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. PMLR, 2014.

[9] Ying Sun, Gesualdo Scutari, and Amir Daneshmand. Distributed optimization based on gradient tracking revisited: Enhancing convergence rate via surrogation. *SIAM Journal on Optimization*, 32(2):354–385, 2022.

[10] Ye Tian, Gesualdo Scutari, Tianyu Cao, and Alexander Gasnikov. Acceleration in distributed optimization under similarity. In *International Conference on Artificial Intelligence and Statistics*, pages 5721–5756. PMLR, 2022.