
OPTIMAL DATA SPLITTING IN DISTRIBUTED OPTIMIZATION FOR MACHINE LEARNING

TECHNICAL REPORT

Gleb Molodtsov

Department of Informatics and Applied Maths
Moscow Institute of Physics and Technology
Moscow, Russia
molodtsov.gl@phystech.edu

Daniil Medyakov

Department of Informatics and Applied Maths
Moscow Institute of Physics and Technology
Moscow, Russia
mediakov.do@phystech.edu

ABSTRACT

Distribution of data among different devices taking into account their performance is considered in this article. The network consists of agents forming a star topology. The goal of this work is to get the most favorable ratio for distributed data to the server and local machines. The optimal gradient sliding algorithm and its application to distributed optimization with similarity are used to solve this problem. The system's running times are compared between uniform and optimal distributions. The higher theoretical performance of our solutions is experimentally confirmed.

1 Introduction

The article [2] considers the problem of finding an optimal algorithm for distributed optimization. Consider the features of the formulation of equation (2) in [2]. The network consists of m agents forming a star topology. The first agent is the main node, and the others are connected only to it and are necessary for additional computational power. We are investigating a problem in which we need to minimize the average loss function over some data, and we want to distribute the solution to the objects in our network, where calculations are performed using the edge nodes of the star topology, and copies of all optimization variables and the connection between the cells are made through the main (central) node. For each unit of the network, a loss function is calculated that measures the difference between the parameter vector and the sample vector. Thus, this problem has been reduced to minimizing the following function:

$$r(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (1)$$

Consider the pros and cons of this formulation of the problem. The advantage of 1 is that by applying the gradient descent algorithm, we can solve our minimization problem for an L -smooth and μ -strongly-convex function r with complexity $\mathcal{O}(\sqrt{\frac{L}{\mu}})$. This upper bound includes a sum of estimates of local calculations on the edge nodes and their communications with the main node. For some classes of problems (where $k = \frac{L}{\mu}$ is not a large number), this method is acceptable. The disadvantage is that for very large k , the algorithm is inefficient. This happens, for example, with high communication costs.

To avoid this cons we transform the function r to the following form:

$$r(x) = f_1(x) + \frac{1}{n} \sum_{i=1}^n [f_i(x) - f_1(x)] \quad (2)$$

For any i , f_i is a convex function as a loss function of local calculations on an auxiliary node, and $f_i - f_1$ is not convex, but contains communication costs. Thus, we have the task of minimizing an L -smooth and μ -strongly-convex function r as the sum of convex and non-convex functions.

For this problem, a modification of the gradient sliding algorithm has been proposed [2], which skips the gradient computation of one function from time to time. All existing gradient sliding algorithms were inapplicable to the problem posed, as they required the convexity of both functions and did not provide optimal upper complexity bounds for both local gradient computation and communication costs. However, this modification of the algorithm fills in the gaps.

2 Problem Statement

Let's consider the accelerated extragradient algorithm (Algorithm 1 in [2]). 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. Let us 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:

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

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 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: $\sum_{i=1}^n b_i = N$, 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 [2]). We obtained the following optimization problem:

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

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 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: $\sum_{i=1}^n b_i = N$, where N is the size of the whole dataset, $\delta = \frac{L}{\sqrt{b_i}}$ (this estimate is given in [2]). We got the following optimization problem:

$$\min_{\sum_{i=1}^n b_i = N; \delta = \frac{L}{\sqrt{b_i}}} [2 \cdot \max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n) \cdot K + 2 \cdot K \cdot \tau_{comm} + \tau_1 \cdot k_{some}] \quad (5)$$

3 Problem solution 5, case $\delta = \frac{L}{\sqrt{b_i}}$

3.1 The primary problem of minimization

In [2] 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, our minimization problem is reduced to:

$$\begin{aligned} \min_{\sum_{i=1}^n b_i = N; \delta = \frac{L}{\sqrt{b_i}}} & [(\max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n) + \tau_{comm}) * \mathcal{O}(\max\{1, \sqrt{\frac{L_p}{\mu}}\} \log(\frac{1}{\varepsilon})) \\ & + \mathcal{O}(\max\{1, \sqrt{\frac{L_q}{L_p}}, \sqrt{\frac{L_p}{\mu}}, \sqrt{\frac{L_q}{\mu}}\} \log(\frac{1}{\varepsilon}))] \end{aligned} \quad (6)$$

3.2 Auxiliary problem

Consider an auxiliary problem:

$$\min_{\sum_{i=1}^n b_i = N; \delta = \frac{L}{\sqrt{b_i}}} [\max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n)] \quad (7)$$

Put $\tau_1^{loc} \leq \tau_2^{loc} \leq \dots \leq \tau_n^{loc}, b_1 \leq b_2 \leq \dots \leq b_n$. And the partition $\{b_i\}_{i=1}^n$ is arbitrary. We renumber b_i so that a smaller τ_i^{loc} corresponds to a larger b_i (otherwise, the solution 7 is clearly not optimal, since renumbering will give a smaller value). Minimize $\max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n)$. Suppose that the largest number must be $\tau_1^{loc} \cdot b_1$. In this case, the partitioning may be such that the first node processes too much information (due to arbitrary partitioning $\{b_i\}_{i=1}^n$) and the solution 7 will not be optimal. Then we get rid of the maximum at node 1 by making partitions $\{b_i\}$ until we get the maximum at the k -th ($k > 1$) node. This situation may again tell us that the part of the dataset $\{b_i\}_{i=k}^n - b_k$ has too much information compared to the rest and the solution 7 is not optimal. Then we will generate part of the dataset $\{b_i\}_{i=k}^n$ until we get the maximum at the j -th ($j > k$) node, and so on until we get the maximum at the n th node. But then b_i may be about the same, and due to the fact that τ_n^{loc} takes a large value, the solution will again be optimal. Thus, we obtained that if the maximum is reached at either the 1st or the n th node, then the solution 7 is not minimal. In the first case, because the distribution of b_i is too unequal, and in the second case, because the value of τ_n^{loc} is large. To avoid these problems, assume that the maximum is reached simultaneously at the 1st and n th nodes. Assume $(\tau_1^{loc} \cdot b_1 = \tau_n^{loc} \cdot b_n) \leq (\tau_2^{loc} \cdot b_2 = \tau_{n-1}^{loc} \cdot b_{n-1}) \leq \dots$. Then $\max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_n^{loc} \cdot b_n) = \max(\tau_1^{loc} \cdot b_1, \tau_2^{loc} \cdot b_2, \dots, \tau_{\lceil \frac{n}{2} \rceil}^{loc} \cdot b_{\lceil \frac{n}{2} \rceil})$. Similarly, we assume that the maximum is reached at the 1st and $\lceil \frac{n}{2} \rceil$ -th nodes, $(\tau_1^{loc} \cdot b_1 = \tau_{\lceil \frac{n}{2} \rceil}^{loc} \cdot b_{\lceil \frac{n}{2} \rceil}) \leq (\tau_2^{loc} \cdot b_2 = \tau_{\lceil \frac{n}{2} \rceil - 1}^{loc} \cdot b_{\lceil \frac{n}{2} \rceil - 1}) \leq \dots$. Continuing in this way, we obtain that $\tau_1 \cdot b_1 = \tau_2 \cdot b_2 = \dots = \tau_n \cdot b_n$. In other words, the dataset must be divided proportionally.

Let us return to the problem 6. In addition to the minimum expression already studied in the problem 7, there are additional terms in the problem 6. They depend on the value of b_1 , but do not depend on $b_i, i \in \overline{2, n}$. Note that $L_q = L, L_p = \delta = \frac{L}{\sqrt{b_i}}$ or $L_p = \delta = \frac{L}{b_i}$ (this estimate is given in [2]).

3.3 Define the final minimization problem

It follows from the previous reasoning that $b_i \tau_i = \text{const} \forall i \in \overline{2, n}$ due to the symmetry of the problem. Therefore,

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

3.3.1 $\delta = \frac{1}{b_1}$

In the first case the following relations are fulfilled:

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

Then the problem will take the following form:

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

Consider the final form of the minimization problem:

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

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

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

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

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

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

$$3.3.2 \quad \delta = \frac{1}{\sqrt{b_1}}$$

Let us perform similar transformations for this case. Our relations turn into:

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

And the main problem will take the following form:

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

Then write down the final minimization problem:

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

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

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

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

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

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

Let's find the derivatives of the functions:

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

It is impossible to write out the solution of these equations in analytical form, because of their degrees, so we will consider the limiting cases.

3.4 Final solution in limiting cases

3.4.1 $\delta = \frac{1}{b_1}$

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

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

Then on the condition that

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

We get a solution:

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

3.4.2 $\delta = \frac{1}{\sqrt{b_1}}$

Let us find an analytical solution to the problem in two particular cases:

1. $\forall i \hookrightarrow \tau_{comm} \ll \tau_i^{loc}, \forall i \neq j \hookrightarrow \tau_i^{loc} \neq \tau_j^{loc}$
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) $0 < b_1 \leq b_1^0$

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

Consider

$$\tau_1^{loc} < \tau_2^{loc} < \dots < \tau_n^{loc} \tag{12}$$

$$\begin{aligned} (\sum_{i=2}^n \frac{1}{\tau_i^{loc}})^{-1} &= \frac{1}{\frac{1}{\tau_1^{loc}} + \dots + \frac{1}{\tau_n^{loc}}} = \frac{\tau_2^{loc} \cdot \dots \cdot \tau_n^{loc}}{\tau_3^{loc} \cdot \dots \cdot \tau_n^{loc} + \tau_2^{loc} \cdot \tau_4^{loc} \cdot \dots \cdot \tau_n^{loc} + \dots + \tau_2^{loc} \cdot \dots \cdot \tau_{n-1}^{loc}} \\ &\geq \frac{\tau_2^{loc}}{12} \frac{1}{n-1} \gg \tau_{comm} \end{aligned} \tag{13}$$

Then taking into account 13:

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

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

The analytical solution in this case is not given, but we note that b_1 should be as large as possible, that is, $b_1 \rightarrow b_1^0$ when the main server capacity is large, that is, when the server capacity is small τ_1^{loc} .

b) $b_1^0 \leq b_1 \leq N$

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

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

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

$$\text{taken at } b_1 = b_1^0 = \frac{N \left(\sum_{i=2}^n \frac{1}{\tau_i^{loc}} \right)^{-1}}{\tau_1^{loc} + \left(\sum_{i=2}^n \frac{1}{\tau_i^{loc}} \right)^{-1}}$$

Thus, in the case of small τ_{comm} we obtained the following result:

$$b_{min} \rightarrow b_1^0 = \frac{N \left(\sum_{i=2}^n \frac{1}{\tau_i^{loc}} \right)^{-1}}{\tau_1^{loc} + \left(\sum_{i=2}^n \frac{1}{\tau_i^{loc}} \right)^{-1}}. \text{ And with increasing server power } b_{min} \text{ becomes almost equal to } b_1^0.$$

Consider case 2.

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

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

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

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

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

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

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

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:

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

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

3.5 Practical solution

In order to practically find the minimum of these functions on the corresponding half-intervals, we will search for points where the derivatives of $\mathcal{F}'_1(b_1)$ and $\mathcal{F}'_2(b_1)$ are converging to zero. Note that due to the type of these functions, their derivatives are zero at most once on the desired half-interval. Then, using the simplest methods, we can find the zeros of the derivatives. After finding these points, we should compare the value of the corresponding function in them with the value at the extreme point of the interval. One of these points will yield the minimum solution. This will be the final solution of 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":

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

The file a9a.txt with number of lines $N = 97683$ was chosen as dataset. The first step was the implementation of algorithm 1 of [2].

In the first stages $\arg \min_x [p(x_k^g) + \langle \nabla p(x_k^g), x - x_k^g \rangle + \frac{1}{2\Theta} \|x - x_k^g\|^2 + q(x)]$ was searched explicitly (Line 5 of Algorithm 1). This was done by equating the gradient to zero. Applying to the problem 18:

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

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

From the obtained convergence graph we found the number of iterations to achieve the given accuracy, 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.

As a result, no time acceleration was obtained. This is due to the peculiarities of the implementation. The code was written in Python, and, as noted above, the expression from line 5 of Algorithm 1 was found analytically. However, due to the way this programming language works, the time required to perform operations involving multiplication of matrices was short in comparison with the time required to perform normal arithmetic operations. As a consequence, the desired effect did not occur.

Then the iterative OGM-G method of [1] was applied to find the solution of line 5 of Algorithm 1. After calculating the required number of iterations to achieve a certain accuracy, the running time was measured.

The updated Algorithm 1 with OGM-G applied was also run on the obtained and uniform data distributions.

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 (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 1

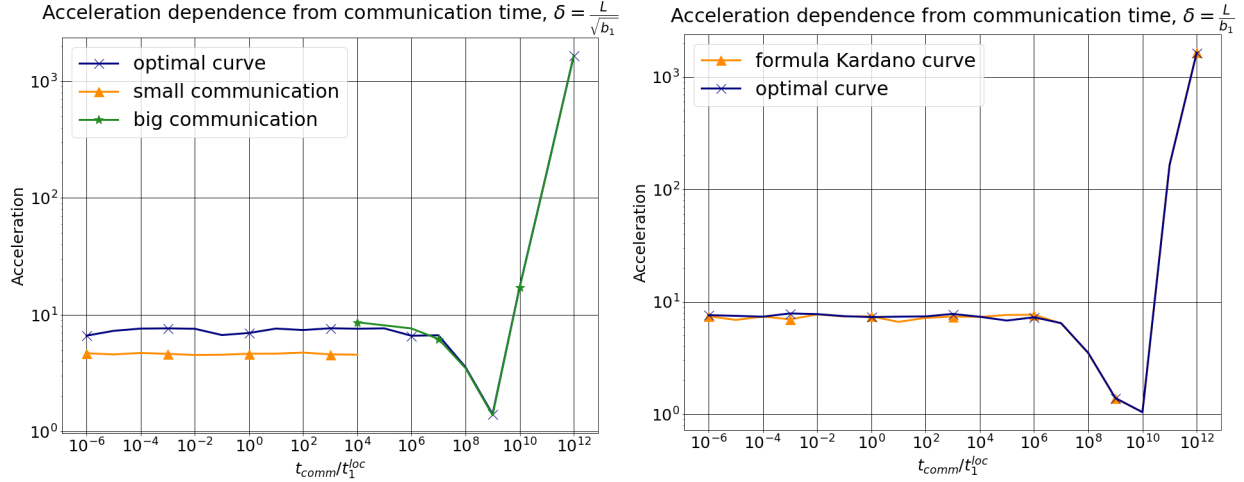


Figure 1: Final results

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.

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

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

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

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

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

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

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 nonconvex subproblems and applying the accelerated extragradient algorithm from [2] as well as the OGM-G algorithm from [1]. Our method works well on star topology networks with various communication costs between the server and the local machines. The theoretical results have been confirmed experimentally. This indicates that our method gives acceleration on tasks of this type.

References

- [1] 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.
- [2] 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.