Accelerated Methods with Compression for Horizontal and Vertical Federated Learning

Stanko Sergey [1] Karimullin Timur [1] Aleksandr Beznosikov [1,2] Alexander Gasnikov [1]

[1] Moscow Institute of Physics and Technology, Russian Federation
[2] Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates

Abstract

Distributed optimization algorithms have emerged as a superior approaches for solving machine learning problems. To accommodate the diverse ways in which data can be stored across devices, these methods must be adaptable to a wide range of situations. As a result, two orthogonal regimes of distributed algorithms are distinguished: horizontal and vertical. During parallel training, communication between nodes can become a critical bottleneck, particularly for high-dimensional and over-parameterized models. Therefore, it is crucial to enhance current methods with strategies that minimize the amount of data transmitted during training while still achieving a model of similar quality. This paper introduces two accelerated algorithms with various compressors, working in the regime of horizontal and vertical data division. By utilizing a momentum and variance reduction technique from the `Katyusha` algorithm, we were able to achieve acceleration and demonstrate one of the best asymptotics for the horizontal case. Additionally, we provide one of the first theoretical convergence guarantees for the vertical regime. Our experiments involved several compressor operators, including RandK and PermK, and we were able to demonstrate superior practical performance compared to other popular approaches.

1 Introduction

As machine learning continues to gain popularity, it is essential for training algorithms to maintain exceptionally high performance in order to effectively address real-world issues. To increase the quality of models' predictions, larger datasets are used. However, training models on a huge amount of samples on a single machine can be very time-consuming, which is why faster optimization approaches are needed.

To address this problem, the scientific community came to distributed algorithms, where the calculation process is divided among different devices (also called machines, nodes, clusters, or workers). Such parallel computation can be used in situations, where data is distributed across several machines, as in the cases of classical distributed Verbraeken et al. [2020] and federated learning approaches [Konečný et al., 2017]. The latter can be divided into two different regimes. One of these is horizontal federated learning, which is the most popular and extensively studied approach. In this scenario, each worker possesses their own collection of samples, but share the same set of features. A different situation is considered in the vertical case [Zhang et al., 2021], where every device has a unique set of features of the same samples. The practical use of the latter regime can be seen in a simple example, where a digital finance company wants to evaluate the risk of approving a loan. To make a qualified prediction, it needs to collect different types of data (features) from the same person (sample). For example, online shopping information from an E-commerce company or average monthly deposit and account information from a bank.

However, legal restrictions or competition between participants prevent these organizations from sharing data, which is why a vertical case is suitable in this situation [Gu et al., 2020]. But classical distributed and horizontal federative learning methods can not be forgotten, as they can show a great performance in accelerating training speed [Goyal et al., 2018]. This is why distributed optimization is gaining a lot of attention nowadays.

However, parallelization of a task practically does not lead to an ideal decrease in time. That means that having N devices does not accelerate task by N times. This happens because of limited ability of networks to exchange information. Thus, the key bottleneck of parallel computation is the communication part. There have been considered several ways of dealing with this issue Konečný et al. [2016], Smith et al. [2018], but in our paper we concentrate solely on reducing communication cost of each iteration by decreasing the size of sending information also known as compression technique Chilimbi et al. [2014], Alistarh et al. [2017]. Due to reducing quality of communicated information, gradient descent methods perform greater amount of iterations, but overall time complexity can be reduced.

Compression mathematically can be represented in the form of a vector function, which is called compression operator. It takes a vector and returns a vector of the same dimension, but the last usually takes less resources in communication networks. In our paper, we consider such operators with the following properties:

Definition 1.1. We say that the compression operator Q is unbiased, if

$$\mathbb{E}\left[Q(x)\right] = x \ , \forall x \in \mathbb{R}^d.$$

We also assume that there exists a constant $w \geq 1$, such:

$$\mathbb{E}\left[\|Q(x)\|^2\right] \leq \omega \|x\|^2 \ , \forall x \in \mathbb{R}^d.$$

We say that the compressor operator Q has density coefficient β, if $Q(x)$ requires in β times less space complexity than x.

1.1 Related works

Unbiased compression. An idea to shrink the vector was researched in [Nesterov, 2012]. This work developed a variant of single-node gradient descent in which only some random coordinates are updated at every step. Later, in the literature on compression, operators of this kind began to be called "RandK" [Beznosikov et al., 2024]. In the paper by [Richtárik and Takáč, 2013], coordinate descent method was adapted for distributed optimization. Generalization with arbitrary unbiased compressor was firstly introduced in `QSGD` paper [Alistarh et al., 2017], but this method does not converge to the true solution, but rather to some neighbourhood, dependent on functions' variance [Gorbunov, 2021].

Variance reduction. The problem mentioned above is characterized by a non-zero limit in the optimal point of the variance of stochastic gradient. A technique of choosing a vector of step direction to fix this challenge is called variance reduction. Firstly used in standard optimization problem Johnson and Zhang [2013], Nguyen et al. [2017] and later was adapted for distributed optimization in `DIANA` [Mishchenko et al., 2023], by compressing not gradient itself, but rather gradient difference, and later for non-convex distributed case in `MARINA` [Gorbunov et al., 2022]. This technique is also used in variational inequalities, for example, in [Beznosikov et al., 2023].

Acceleration. The optimal algorithm for a non-distributed strongly convex problem, which uses variance reduction technique, is `Katyusha` [Allen-Zhu, 2018]. It is based on acceleration technique, called by authors "negative momentum" with the combination of stochastic adjustment of the gradient in the old point. Vector, in which gradient is calculated, updates with the loop,

Table 1: Summary of bounds for iteration complexities for finding an ε-solution. Convergence is measured by the distance to the solution.

Regime	Reference	Iteration complexity
Horizontal	QSGD Alistarh et al. [2017] [(1)]	$\mathcal{O}\left(\frac{L}{\mu}\left(1+\frac{\omega-1}{n}\right)\log\frac{1}{\varepsilon}\right)$
	DIANA Mishchenko et al. [2023]	$\mathcal{O}\left(\left(\frac{L}{\mu}\left(1+\frac{2(\omega-1)}{n}\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$
	VR-DIANA Horváth et al. [2019]	$\mathcal{O}\left(\left(\frac{L}{\mu}\left(1+\frac{\omega-1}{n}\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$
	ADIANA Li et al. [2020] [(2)]	$\mathcal{O}\left(\left(\sqrt{\frac{L}{\mu}}\left(1+\sqrt{\left(\frac{\omega-1}{n}+\sqrt{\frac{\omega-1}{n}}\right)\omega}\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$
	ADIANA He et al. [2024] [(3)]	$\mathcal{O}\left(\left(\sqrt{\frac{L}{\mu}}\left(\sqrt{\frac{\omega^2}{n}}+1\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$
	MARINA Gorbunov et al. [2022] [(4)]	$\mathcal{O}\left(\left(\frac{L}{\mu}\left(1+\frac{\omega-1}{\sqrt{n}}\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$
	MASHA Beznosikov et al. [2023] [(5)]	$\mathcal{O}\left(\left(\frac{L}{\mu}\sqrt{\left(w+\frac{(\omega-1)^2}{n}\right)}+\omega\right)\log\frac{1}{\varepsilon}\right)$
	Three Pillars Algorithm Beznosikov and Gasnikov [2023] [(5, 6)]	$\mathcal{O}\left(\left(\frac{L}{\mu}\sqrt{n}+n\right)\log\frac{1}{\varepsilon}\right)$
	Algorithm 1(This paper)	$\mathcal{O}\left(\left(\sqrt{\frac{L}{\mu}}\left(\sqrt{\frac{\omega^2}{n}}+1\right)+\omega\right)\log\frac{1}{\varepsilon}\right)$

Vertical	AVFL Cai et al. [2022]; CE-VFL Sun et al. [2023];
	SecureBoost+ Chen et al. [2021]; eHE-SecureBoost Xu et al. [2021];
		No theoretical results

	CVFL Castiglia et al. [2023]	No concrete number of iterations [(7)]
	Algorithm 2(This paper)	$\mathcal{O}\left(\left(\sqrt{\frac{\bar{L}}{\mu}s}+s\right)\log\frac{1}{\varepsilon}\right)$

[(1)] correct tuning of step size, no convergence with fixed step size; [(2)] this is the complexity derived in the original paper Li et al. [2020]; [(3)] this is the complexity derived by a refined analysis in the preprint He et al. [2024] [(4)] under PL condition; [(5)] for VI and SPPs; [(6)] for PermK compressor; [(7)] for special compressor convergence rate is $\mathcal{O}(\frac{1}{\sqrt{K}})$, no guaranties in the case of arbitrary compressor;
Notation: μ = constant of strong convexity, L = smoothness constant of the target function, ω = compression constant (see Definition 1.1), n = number of workers, s = number of samples, $\bar{L} = \frac{1}{s}\sum\limits_{i=1}^{s} L_i$

which is theoretically no more effective than an update with a certain probability represented in L-Katyusha [Kovalev et al., 2019], but is difficult to perceive and empirically slower.

1.2 Our contributions

- **New distributed algorithms.** We present new distributed algorithms with compression for various regimes of data division. As mentioned before, parallel algorithms can be based on non-parallel approaches with variance reduction, e.g.: for non-convex problem, MARINA [Gorbunov et al., 2022] is based on PAGE [Li et al., 2021], or for variational inequalities, MASHA Beznosikov et al. [2023] is based on [Alacaoglu and Malitsky, 2022]. In our paper, we base our results on L-Katyusha [Kovalev et al., 2019], one of the state-of-the-art algorithm for strongly convex problem with variance reduction technique.

  - **Horizontal regime.** We propose L-Katyusha-based algorithm with horizontal data division – DHPL-Katyusha (Distributed Horizontally Partitioned L-Katyusha from Algorithm 1), in which every worker compresses and sends the difference between gradients of its own function in new and old points, after which regular L-Katyusha algorithm is performed locally. The asymptotics of our algorithm is compared with other popular approaches in Table 1. Unlike QSGD [Alistarh et al., 2017], DIANA [Mishchenko et al., 2023], VR-DIANA [Horváth et al., 2019], MASHA [Beznosikov et al., 2023] and Three Pillars Algorithm [Beznosikov and Gasnikov, 2023], our algorithm is accelerated (e.g. has asymptotics proportional to $\sqrt{\frac{L}{\mu}}$, where L is a smoothness constant and μ is a constant of strong convexity), and in comparison with the improved version of ADIANA, DHPL-Katyusha has the same asymptotics, but shows better empirical results in experiments.

  - **Vertical regime.** We strive to present DVPL-Katyusha and DVPL-Katyusha with scalar `compression` (Algorithm 2 and 3 respectively) — algorithms with compression working under the assumption of the vertical data division. The first one uses RandK compressor, and the second utilize MSE loss and arbitrary (with respect to Definition 1.1) compressor. In both algorithms every worker sends its own part of scalar values, needed to calculate total loss, and then perform other part of `L-Katyusha` locally with its own set of features. We estimate the asymptotics in such settings and show a lower bound for scalar compression on a constructive example. We are one of the first to present algorithm with vertical regime and guaranties for theoretical convergence. Other approaches, represented in Table 1, such as `AVFL` [Cai et al., 2022], `CE-VFL` [Sun et al., 2023], `SecureBoost+` [Chen et al., 2021] and `eHE-SecureBoost` [Xu et al., 2021] do not have it. The `CVFL` algorithm [Castiglia et al., 2023] has theoretical convergence rates only for special compressors, but also lacks concrete convergence guarantees.

- **Various compressors.** We obtain similar estimates for different compressors. By obtaining ω from Definition 1.1, we research RandK operator, which retains only K random coordinates from a vector. In this paper, we also implement one of the state-of-the-art compressor PermK [Szlendak et al., 2021], which distributes random permutations of vector components across all workers to compute only on them. As PermK can be represented in the form of several correlated compressors, ω can not be defined for it. Thus, a slightly different analysis is used in order to develop a proof.

- **Numerical experiments.** In numerical experiments, we illustrate the most important properties of the new methods. The results correspond to the theory developed.

1.3 Technical preliminaries

Notations. $\langle x, y\rangle := \sum_{i=1}^{d} x_i y_i$ is used to denote standard inner product of $x, y \in \mathbb{R}^d$. Operator $\mathbb{E}[\cdot]$ denotes mathematical expectation. We denote $A_i^T \in \mathbb{R}^d$ as the i-th column of the matrix A^T.

2 Horizontal Case

In this section, we provide an algorithm for the horizontal division of data, which we call `DHPL-Katyusha` (Algorithm 1). We state the standard distributed learning problem: each worker has its own dataset, on which it can calculate the loss. The target loss function, which we need to minimize, is obtained by summarizing and averaging all workers' losses. Formally, it can be written in the form of:

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]. \tag{1}$$

Communications between nodes can be centralized, e.g. they can be managed only by the server, which in gradient approaches collects functions' gradients, averages them, does subsequent calculation to find a new point and broadcasts it to other devices. Such setting is considered to be popular, especially in theory. Another way of organizing communications is a decentralized setup, where all devices are connected in a single network with specific topology and, unlike centralized, can communicate with each other through networks' edges. In our work, we prefer the latter one.

The situation in which each worker's compressor is completely independent of the compressor operators of other devices seems to be rather improper for the decentralized setup. There are several reasons for such difficulties arising, the first is that with such approach we are not able to exploit data similarity between devices to the fullest extent. Explanation is simple, the difference between the averaging of compressed gradients and their uncompressed versions is expressed by the AB-inequality described in [Szlendak et al., 2021]. For uncorrelated compressors, the value of constant B is expressed as zero, while for correlated compressors it is significantly higher, which allows writing tighter estimates. The second problem we face lies on the level of technical implementation of communications in a network, the size of the transmitted packets in which is unknown in advance. Thus, it becomes complicated to perform such important operations as AllReduce [Chan et al., 2007], which is widely used in the present work, using arbitrary compressors. This is why correlated compressors, like PermK [Szlendak et al., 2021], are used in communication networks. In our paper we research both correlated and uncorrelated compressors, therefore to make algorithm suitable for different ways of organizing communications we change the name of specific communication operation with the word "broadcast".

2.1 Introduction of `DHPL-Katyusha`

At the beginning of the k-th iteration of `DHPL-Katyusha`, every worker computes x^k locally, as a convex combination of z^k, w^k and y^k and broadcasts it to all other workers (line 3). After this, each worker already has the same old and new point and therefore can compute the difference between local gradients in the current point x^k and in the old point w^k in the same manner (line 4). Then, each worker applies a local compressor and sends this difference to everyone else (line 5, line 6). Having done that, every device is able to find the average of all compressed differences and add full gradient in w^k (stored locally on each device) to apply the variance reduction technique (line 7). After that, "negative momentum" idea is used from [Kovalev et al., 2019] (line 8, line 9). Finally, w^k is updated to the actual point, based on a coin flip, which all workers do with the same random seed (line 10). If w^k is adjusted, the uncompressed gradient is calculated by AllReduce procedure (line 13). As p strives to zero with $\omega \to \infty$, full gradient is updated rarely.

Note that the Q_i compressors in this algorithm work independently of each other, only if the PermK operator is not considered.

2.2 Convergence results

To prove the asymptotics of `DHPL-Katyusha` we need to make the following assumptions for the problem (1).

A 1. Functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are L-smooth for some $L > 0, \forall i \in \overline{1,n}$:
$f_i(y) \leq f_i(x) + \langle \nabla f_i(x), y - x\rangle + \frac{L}{2}\|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$.

Note that with such assumption, the target function $f(x)$ is also L - smooth.

A 2. The function $f : \mathbb{R}^d \to \mathbb{R}$ is μ - strongly convex for some $\mu > 0$:
$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$.

In the original paper [Kovalev et al., 2019], one of the steps in the proof of the asymptotics of `L-Katyusha`, which is called as Lemma 6, is the estimation of a term $\|g^k - \nabla f(x^k)\|^2$. It can be formally written in the following form:

Lemma 1. For the original `L-Katyusha` algorithm, the norm of the difference between the true and the real gradient can be estimated as

$$\|g^k - \nabla f(x^k)\|^2 \leq 2L \left(f(w^k) - f(x^k) - \langle \nabla f(x^k), w^k - x^k\rangle \right).$$

However, in our paper, g^k is changed to $\widetilde{g}^k$, and therefore, it takes to utilize additional inequalities to gain similar terms for an upper estimate of $\|\widetilde{g}^k - \nabla f(x^k)\|^2$.

Algorithm 1 `DHPL-Katyusha`

Input: initial $y^0 = w^0 = z^0 \in \mathbb{R}^d$, step size $\eta = \min\left\{\frac{\theta_2}{(1+\theta_2)\theta_1}, \frac{\frac{\widetilde{L}}{L}\theta_2}{(1+\theta_2)\theta_1}\right\}$, where $\widetilde{L} = L\frac{\omega}{n}$, $\sigma = \frac{\mu}{\widetilde{L}}$, parameters $\theta_1, \theta_2 \in \mathbb{R}$ and probability $p \in (0, 1]$ (every worker has the same random seed for calculating p).

```
for $k = 0, 1, 2, \dots K$ do
    for $i = 1 \dots n$ in parallel do
        $x^k \leftarrow \theta_1 z^k + \theta_2 w^k + (1 - \theta_1 - \theta_2)y^k$
        $g_i^k \leftarrow \nabla f_i(x^k) - \nabla f_i(w^k)$
        $\widetilde{g}_i^k \leftarrow Q_i(g_i^k)$
        Using communications broadcast $\widetilde{g}_i^k$
        Compute $\widetilde{g}^k \leftarrow \frac{1}{n}\sum_{i=1}^{n} \widetilde{g}_i^k + \nabla f(w^k)$
        $z^{k+1} \leftarrow \frac{1}{1+\eta\sigma}(\eta\sigma x^k + z^k - \frac{\eta}{\widetilde{L}}\widetilde{g}^k)$
        $y^{k+1} \leftarrow x^k + \theta_1(z^{k+1} - z^k)$
        $w^{k+1} \leftarrow \begin{cases} y^k, & \text{with probability } p \\ w^k, & \text{with probability } 1-p \end{cases}$
        if $w^{k+1} = y^k$ then
            Using communications broadcast $\nabla f_i(w^{k+1})$
            Compute $\nabla f(w^{k+1}) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^{k+1})$
        end if
    end for
end for
```

Lemma 2. For the `DHPL-Katyusha` algorithm, the norm of the difference between the true and the real gradient can be estimated as

$$\|\widetilde{g}^k - \nabla f(x^k)\|^2 \leq 2\widetilde{L}\left(f(w^k) - f(x^k) - \langle \nabla f(x^k), w^k - x^k \rangle\right),$$

where

$$\widetilde{L} = L\frac{\omega}{n}.$$

As similar inequality is proven for several algorithms, we give the following definition:

Definition 2.1. In Katyusha-based algorithm we call $\widetilde{L}$ the efficient Lipschitz constant, if it is the smallest constant, such that:

$$\|\widetilde{g}^k - \nabla f(x^k)\|^2 \leq 2\widetilde{L}\left(f(w^k) - f(x^k) - \langle \nabla f(x^k), w^k - x^k \rangle\right).$$

Note, that for the case $\widetilde{L} \geq L$ we can use all Lipschitz inequalities that were used in [Kovalev et al., 2019], but for $\widetilde{L}$ and therefore conclude a proof. But not always this property is satisfied, and we need to use another inequalities to conclude a proof. Using this fact, we can formulate the convergence theorem:

Theorem 1. Let Assumptions 1, 2 be hold. Denote $\widetilde{L}$ as $L\frac{\omega}{n}$ and x^* as the solution for the problem 1. Then after k iterations of `DHPL-Katyusha`

$$\begin{aligned}&\mathbb{E}\left[\mathbb{Z}^{k+1} + \mathbb{Y}^{k+1} + \mathbb{W}^{k+1}\right] \\ &\leq \frac{1}{1+\eta\sigma}\mathbb{Z}^k + (1-\theta_1(1{-}\theta_2))\mathbb{Y}^k + \left(1 - \frac{p\theta_1}{1+\theta_1}\right)\mathbb{W}^k,\end{aligned}$$

where:

$$\begin{aligned}\mathbb{Z}^k &:= \frac{\widetilde{L}(1+\eta\sigma)}{2\eta}\|z^k - x^*\|^2, \\ \mathbb{Y}^k &:= \frac{1}{\theta_1}\left(f(y^k) - f(x^*)\right), \\ \mathbb{W}^k &:= \frac{\theta_2(1+\theta_1)}{p\theta_1}\left(f(w^k) - f(x^*)\right).\end{aligned}$$

We choose the concrete p with the aim of reducing the average amount of information sent in each communication procedure. At each iteration, on average, workers communicate at a cost of $\mathcal{O}\left(\frac{1}{\beta} + p\right)$. Therefore, p should be equal to $\frac{1}{\beta}$ to get a gain in asymptotics. Other parameters we choose are similar to that of `L-Katyusha` [Kovalev et al., 2019].

Corollary 1. Denote β as a density coefficient of compressor operator Q from Definition 1.1. Let $p = \frac{1}{\beta}$, $\theta_1 = \min\{\sqrt{\frac{2\sigma\beta}{3}}, \frac{1}{2}\}$, $\theta_2 = \frac{1}{2}$. Then after $K = \mathcal{O}\left(\left(\sqrt{\frac{L}{\mu}}\left(\sqrt{\beta\frac{\omega}{n}} + 1\right) + \beta\right)\log\frac{1}{\varepsilon}\right)$ iterations $\mathbb{E}\left[\Psi^K\right] \leq \varepsilon\Psi^0$, where, the Lyapunov function Ψ^K is defined as $\Psi^K := \mathbb{Z}^K + \mathbb{Y}^K + \mathbb{W}^K$. Total information, sent by `DHPL-Katyusha` with such parameters, is

$$\mathcal{O}\left(\left(\sqrt{\frac{L}{\mu}}\left(\sqrt{\frac{1}{\beta}\frac{\omega}{n}} + 1\right) + 1\right)\log\frac{1}{\varepsilon}\right).$$

2.3 RandK and PermK comparison

For RandK compressor, the exact ω is equal to $\frac{d}{K}$ [Beznosikov et al., 2024]. Therefore, using Lemma 2 we get the efficient Lipschitz constant, which is always greater than L:

Lemma 3. (The efficient Lipschitz for RandK compressor in the horizontal case)
For `DHPL-Katyusha` with RandK compressor, the efficient Lipschitz constant is less than:

$$\widetilde{L}_{Rand} = \left(\frac{d}{nK} + 1\right)L$$

As Q_i for PermK operator are correlated, the analysis for it is slightly different from the original `DHPL-Katyusha`. But we still managed to find its efficient Lipschitz constant and therefore can estimate asymptotics for it:

Lemma 4. (The efficient Lipschitz for PermK compressor in the horizontal case)
For `DHPL-Katyusha` with PermK compressor, the efficient Lipschitz constant is equal to L.

As it can be seen in the above lemmas, $\widetilde{L}_{Rand}$ and $\widetilde{L}_{Perm}$ are asymptotically equal as we take $K = \frac{d}{n}$ for RandK in order to compare with PermK. However, as it is shown in the experiments, PermK compressor is empirically better for horizontal data division case than RandK.

3 Vertical Case

In this section, we introduce the Katyusha-based algorithm of distributed optimization with vertical data division, which is named as DVPL-Katyusha. We introduce an algorithm for the problem of minimization of the linear loss function. This can be formally written as:

$$\min_{x\in\mathbb{R}^d} \left[\mathcal{L}(Ax, b) := \frac{1}{s} \sum_{j=1}^{s} \mathcal{L}_j(A_j^T x, b_j) \right], \tag{2}$$

where $b \in \mathbb{R}^s$ is a vector of targets, $A \in \mathbb{R}^{s\times d}$ is a feature matrix, s is a number of samples and d is a number of features. We denote A_j as a j-th row of matrix A.

In the vertical setup, we assume that every worker has its own set of features, which is formally represented, as a set of columns of the feature matrix A. Thus, we can rewrite the dot product $A_j^T x$ in the form of:

$$A_j^T x = \sum_{i=1}^{n} A_{ji}^T x_i, \tag{3}$$

where as A_i and x_i we denote submatrix and point subvector corresponding to the set of i-th worker's features, n is a number of devices.

Compressor in DVPL-Katyusha is basically a RandK compressor Beznosikov et al. [2024], which originally compresses data by choosing random coordinates with certain probabilities and broadcasts information only from them. The main difference in our case if that every worker has the same random seed for such choice, therefore it can be viewed as an analogue to batching technique for the vertical case.

3.1 Introduction of DVPL-Katyusha

In DVPL-Katyusha every worker has its own set of features. Therefore, unlike the horizontal regime, in the vertical case all operations are performed on subvectors, corresponding to individual worker's components. We denote such subvectors with the same letter as the vector itself, but with lower index i. Similar to DHPL-Katyusha (Algorithm 1) at each iteration every worker sets x_i^k as a convex combination of z_i^k, w_i^k and y_i^k (line 3). A gradient calculated on the j-th sample depends on the dot product of A_j^T and a point in which we want to perform a computation. This is why, to perform a step, we need to obtain dot products $\langle A_j^T, x^k\rangle$ and $\langle A_j^T, w^k\rangle$ for every sample, chosen by RandK. We want to select samples with large Lipschitz constant more often, therefore, we select each sample with probability $\frac{L_j}{sL}$. As every worker can compute only its own part of this sum (in our denotations for a single sample j it is $\langle A_{ji}^T, x_i^k\rangle$ or $\langle A_{ji}^T, w_i^k\rangle$), all such parts from every chosen sample should be broadcasted in order to compute g^k (line 4, line 6). After this, a standard momentum part of Katyusha is performed locally (line 9, line 10), after which w^k is updated with some probability, with every worker having the same random seed for it (line 11). If w^k is adjusted to the new point, the full gradient (e.g., all samples are used) in this vector is calculated the same way as is written above, but without any compression (line 14). Note that similar to the horizontal case, we consider probability of such update low, e.g., full gradient is calculated infrequently.

3.2 Convergence results

As in DHPL-Katyusha we need to assume our target function to be strong convex and worker's functions to be L-smooth.

Algorithm 2 `DVPL-Katyusha`

Input: initial $y^0 = w^0 = z^0 \in \mathbb{R}^d$, step size $\eta = \frac{\theta_2}{(1+\theta_2)\theta_1}$, $\sigma = \frac{K\mu}{\bar{L}}$ if $\frac{\bar{L}}{K} \geq L$ or else $\sigma = \frac{\mu}{L}$, parameters $\theta_1, \theta_2 \in \mathbb{R}$ and probability $p \in (0,1]$, every worker has the same random seed for RandK random. RandK select j-th sample with probability $p_j = \frac{L_j}{s\bar{L}}$.

```
for $k = 0, 1, 2, \dots K$ do
    for $i = 1 \dots n$ in parallel do
        $x_i^k \leftarrow \theta_1 z_i^k + \theta_2 w_i^k + (1 - \theta_1 - \theta_2) y_i^k$
        Compute $X_i^k = \text{RandK}\left( \left\| \left\langle A_{ji}^T, x_i^k \right\rangle \right\|_{j=\overline{1,s}} \right)$
        Compute $W_i^k = \text{RandK}\left( \left\| \left\langle A_{ji}^T, w_i^k \right\rangle \right\|_{j=\overline{1,s}} \right)$
        Using communications broadcast $X_i^k$ and $W_i^k$
        $J^k = \{j_1^k, \cdots, j_n^k\}$ - indices, selected by RandK
        $g_i^k \leftarrow \frac{1}{K} \sum\limits_{j \in J^k} \frac{1}{sp_j} \nabla \mathcal{L}_j \left( \sum\limits_{i=1}^n X_{ij}^k, b_j \right)_i - \frac{1}{K} \sum\limits_{j \in J^k} \frac{1}{sp_j} \nabla \mathcal{L}_j \left( \sum\limits_{i=1}^n W_{ij}^k, b_j \right)_i + \nabla \mathcal{L}\left( Aw^k, b \right)_i$
        $z_i^{k+1} \leftarrow \frac{1}{1+\eta\sigma}(\eta\sigma x_i^k + z_i^k - \frac{\eta}{\widetilde{L}} g_i^k)$
        $y_i^{k+1} \leftarrow x_i^k + \theta_1(z_i^{k+1} - z_i^k)$
        $w_i^{k+1} \leftarrow \begin{cases} y_i^k, & \text{with probability } p \\ w_i^k, & \text{with probability } 1-p \end{cases}$
        if $w_i^{k+1} = y_i^k$ then
            for $j = 1 \dots s$ do
                Compute $\left\langle A_{ji}^T, w_i^k \right\rangle$
                Using communications broadcast $\left\langle A_{ji}^T, w_i^k \right\rangle$
            end for
            Compute $\nabla \mathcal{L}\left( Aw^k, b \right)_i$
        end if
    end for
end for
```

A 3. Functions $\mathcal{L}_i : \mathbb{R}^d \to \mathbb{R}$ are L-smooth for some $L > 0, \forall i \in \overline{1, s}$:
$\mathcal{L}_i(y) \leq \mathcal{L}_i(x) + \langle \nabla \mathcal{L}_i(x), y - x\rangle + \frac{L}{2}\|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$.

A 4. The function $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is μ - strongly convex for some $\mu > 0$:
$\mathcal{L}(y) \geq \mathcal{L}(x) + \langle \nabla \mathcal{L}(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2$, $\forall x, y \in \mathbb{R}^d$.

To estimate the asymptotics of the algorithm, we need to get a bound of the efficient Lipschitz constant for `DVPL-Katyusha`. Denote $\bar{L}$ as $\frac{1}{s}\sum\limits_{i=1}^{s} L_i$. Therefore, we can formulate the following lemma.

Lemma 5. The efficient Lipschitz constant for `DVPL-Katyusha` is less than $\widetilde{L}$, where:

$$\widetilde{L} = \max\left\{L, \frac{\bar{L}}{K}\right\}.$$

Knowing the efficient Lipschitz constant, we can estimate a convergence rate of our algorithm:

Theorem 2. Let assumptions 3 and 4 be hold. Denote $\widetilde{L}$ as $\max\left\{L, \frac{\bar{L}}{K}\right\}$ and x^* as the solution for the problem 2. Then after k iterations of `DVPL-Katyusha`

$$\mathbb{E}\left[\mathbb{Z}^{k+1} + \mathbb{Y}^{k+1} + \mathbb{W}^{k+1}\right] \leq \frac{1}{1 + \eta\sigma}\mathbb{Z}^k + (1 - \theta_1(1 - \theta_2))\mathbb{Y}^k + \left(1 - \frac{p\theta_1}{1 + \theta_1}\right)\mathbb{W}^k,$$

where:

$$\mathbb{Z}^k := \frac{\widetilde{L}(1 + \eta\sigma)}{2\eta}\|z^k - x^*\|^2,$$
$$\mathbb{Y}^k := \frac{1}{\theta_1}\left(f(y^k) - f(x^*)\right),$$
$$\mathbb{W}^k := \frac{\theta_2(1 + \theta_1)}{p\theta_1}\left(f(w^k) - f(x^*)\right).$$

After choosing concrete parameters, we can get an estimation for a total number of iterations and sent information.

Corollary 2. Let $p = \frac{K}{s}$, $\theta_1 = \min\{\sqrt{\frac{2\sigma s K}{3}}, \frac{1}{2}\}$, $\theta_2 = \frac{1}{2}$. If $\frac{\bar{L}}{K} \geq L$, then after $N = \mathcal{O}\left(\left(\sqrt{\frac{s\bar{L}}{\mu}} + s\right)\log\frac{1}{\varepsilon}\right)$ iterations $\mathbb{E}\left[\Psi^N\right] \leq \varepsilon\Psi^0$, where the Lyapunov function Ψ^N is defined as $\Psi^N := \mathbb{Z}^N + \mathbb{Y}^N + \mathbb{W}^N$. If $L > \frac{\bar{L}}{K}$, then after $\mathcal{O}\left(\left(\sqrt{\frac{sKL}{\mu}} + s\right)\log\frac{1}{\varepsilon}\right)$ iterations, the same accuracy is achieved.

The average information cost of a single iteration is $\mathcal{O}(K + ps)$. Therefore, we choose p as $\frac{K}{s}$ to asymptotically reduce it.

3.3 Scalar compressors

This section introduces `DVPL-Katyusha with scalar compression`, which is the generalization of Algorithm 2 with the compressors that operate on the workers' parts of the dot product

$A_{ji}^T x_i^k$. For simplicity, only the MSE loss function will be considered in this section, which can be formally expressed as:

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{s}||Ax-b||^2\right]. \tag{4}$$

As our setup is still vertical, we can rewrite the above in terms of workers' components:

$$\min_{x\in\mathbb{R}^d}\left[f(x) := \frac{1}{s}\left\|\sum_{i=1}^{n} A_i x_i - b\right\|^2\right]. \tag{5}$$

The practical utility of the MSE loss function is significant, as linear models are quick to train and can serve as effective benchmarks. Additionally, large models such as neural networks can leverage a trained model to generate features and then utilize only the final linear layer. Furthermore, implementing privacy mechanisms in vertical federated learning is simpler with this approach through linear models, like in [Huang et al., 2022].

Like in Algorithm 2 we need to assume L-smoothness and μ strong convexity of our problem. In the MSE loss, this can be written in terms of eigenvalues.

A 5. The matrix $A^T A$ has all eigenvalues bounded in a segment of $[\mu, L]$.

To estimate the asymptotics of the algorithm, we need to get a bound of the efficient Lipschitz constant for `DVPL-Katyusha with scalar compression`.

Lemma 6. The efficient Lipschitz constant for `DVPL-Katyusha with scalar compression` is less than $\widetilde{L}$, where:

$$\widetilde{L} = L\left(1 + (\omega - 1)\frac{s\cdot\sum\limits_{j=1}^{s} L_j^2}{\mu^2}\right).$$

We also proved that under made assumptions there exists a function that has $\widetilde{L}$ proportional to $\frac{L^3}{\mu^2}$, therefore our asymptotics for `DVPL-Katyusha with scalar compression` is optimal in the terms of L and μ constants.

Lemma 7. There exists such function that holds under 5 that its efficient Lipschitz constant $\widetilde{L}$ is proportional to $(\omega-1)\frac{L^3}{\mu^2}$.

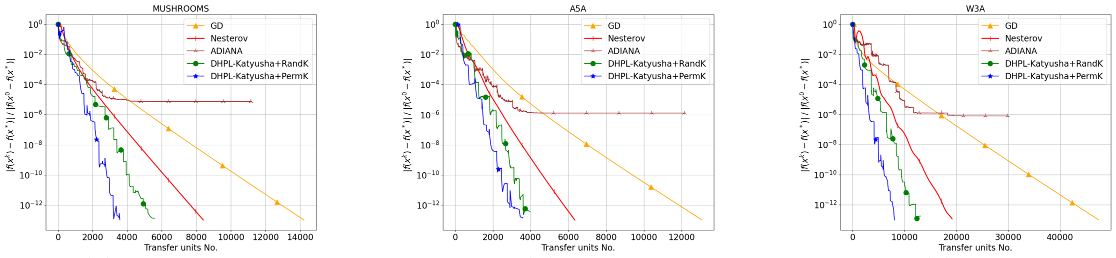
3.4 PermK compressor

Beside RandK and scalar operators, we have also implemented another compression strategy [Szlendak et al., 2021]. In PermK every worker choose its own set of samples, which can not intersect with any other set. After that, each worker do AllReduce with calculated values. Therefore, we can prove a theorem:

Algorithm 3 `DVPL-Katyusha with scalar compression`

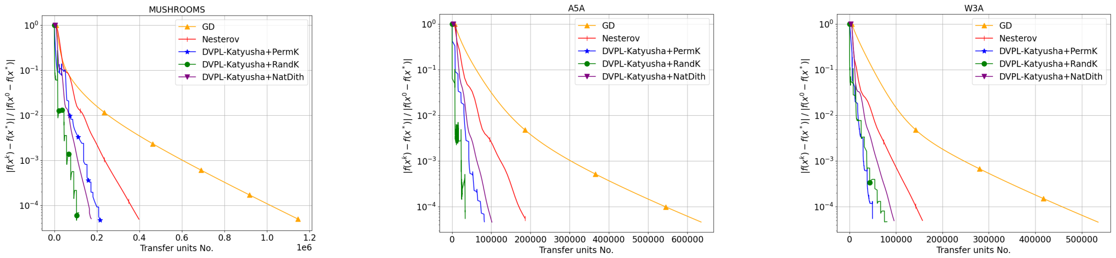
Input: initial $y^0 = w^0 = z^0 \in \mathbb{R}^d$, step size $\eta = \frac{\theta_2}{(1+\theta_2)\theta_1}$, $\sigma = \frac{\mu}{\widetilde{L}}$, where $\widetilde{L} = L\left(1 + (\omega - 1)\frac{s\cdot\sum\limits_{j=1}^{s} L_j^2}{\mu^2}\right)$, parameters $\theta_1, \theta_2 \in \mathbb{R}$ and probability $p \in (0,1]$, every worker has the same random seed for RandK random. RandK select j-th sample with probability $p_j = \frac{1}{s}$.

```
for $k = 0, 1, 2, \dots K$ do
    for $i = 1 \dots n$ in parallel do
        $x_i^k \leftarrow \theta_1 z_i^k + \theta_2 w_i^k + (1 - \theta_1 - \theta_2) y_i^k$
        Compute $D_i^k = \text{RandK}\left(\left\|\left\langle A_{ji}^T, x_i^k - w_i^k \right\rangle\right\|_{j=\overline{1,s}}\right)$
        $J^k = \{j_1^k, \cdots, j_n^k\}$ - indices, selected by RandK
        Using communications broadcast $Q_i\left(D_i^k\right)$
        $g_i^k \leftarrow \frac{2}{b_s} \sum\limits_{j \in J^k} A_{ji}^T \sum\limits_{i=1}^{n} Q_i\left(D_{ij}\right) + \frac{2}{s}\left(A^T A w^k - A^T b\right)_i$
        $z_i^{k+1} \leftarrow \frac{1}{1+\eta\sigma}(\eta\sigma x_i^k + z_i^k - \frac{\eta}{\widetilde{L}} g_i^k)$
        $y_i^{k+1} \leftarrow x_i^k + \theta_1(z_i^{k+1} - z_i^k)$
        $w_i^{k+1} \leftarrow \begin{cases} y_i^k, & \text{with probability } p \\ w_i^k, & \text{with probability } 1 - p \end{cases}$
        if $w_i^{k+1} = y_i^k$ then
            for $j = 1 \dots s$ do
                Compute $\left\langle A_{j_i}^T, x_i^k - w_i^k \right\rangle$
                Using communications broadcast $\left\langle A_{ji}^T, x_i^k - w_i^k \right\rangle$
            end for
            Compute $\frac{2}{s}\left(A^T A w^k - A^T b\right)_i$
        end if
    end for
end for
```



(a) `mushrooms` (b) `a5a` (c) `w3a`

Figure 1: Comparison of different algorithms for solving optimization problem from section 4.1 in horizontal case on LIBSVM datasets `mushrooms`, `a5a` and `w3a`.



(a) `mushrooms` (b) `a5a` (c) `w3a`

Figure 2: Comparison of different algorithms for solving optimization problem from section 4.2 in vertical case on LIBSVM datasets `mushrooms`, `a5a` and `w3a`.

Theorem 3. The efficient Lipschitz constant for PermK compressor is less than:

$$\widetilde{L}_{Perm} = 2L\frac{s\sum\limits_{j=1}^{s} L_j^2}{\mu^2}.$$

Unlike horizontal regime, RandK has a superior theoretical asymptotics on PermK.

4 Numerical Experiments

In this section, we conduct multiple numerical experiments on horizontal and vertical cases with a binary classification problem and a strongly convex target function on `mushrooms`, `a5a` and `w3a` datasets from the LIBSVM library.

4.1 Binary classification in the horizontal regime

As assumed in our paper, in horizontal case we present our target function as the following finite sum:

$$\min_{x\in\mathbb{R}^d}\left\{f(x) = \frac{1}{n}\sum_{m=1}^{n} f_m(x)\right\},$$

where $f_m(x) = \frac{1}{s/n}\sum_{i=1}^{s/n}\log(1+\exp(-y_i a_i^T x)) + \frac{\lambda}{2}\|x\|_2^2$ – the sum here is taken with samples $(a_i,\, y_i)$ from the local dataset of the m-th worker, $\lambda = L/100$ – L here is the target function's $f(x)$ smoothness constant.

In our experiments on horizontal case, we have $n = 100$ devices involved in the training process and compare `DHPL-Katyusha` in combination with Rand1% and PermK compressors with accelerated algorithm `ADIANA` and vanilla GD and AGD, running in the same distributed setting.

All of the plots on the Figure 1 are constructed for the convergence with tuned parameters. It is known from the article Chan et al. [2007] that the AllReduce of vectors of size d takes time proportional to the magnitude of $2\frac{n-1}{n}d$, where d is the size of the vectors transmitted in this way. On the x-axis, we postponed either $\frac{d}{\beta}$ or d, depending on the outcome of the coinflip produced during the iteration. Thus, on the x-axis, we have a number proportional to the number of numpy.doubles (without taking into account the $2\frac{n-1}{n}$ multiplier that is the same for all the plots) transmitted during the entire learning process.

As it was expected in theory, our algorithm has an advantage over AGD with a total communication complexity of $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\right)$ due to the n term in the denominator of our algorithm's best complexity – $\mathcal{O}\left(\sqrt{\frac{L}{n\mu}}\right)$. More than that, PermK clearly performs better than Rand1% in such setup, as can be seen from the graph.

Finally, it worth to be noted that the ADIANA algorithm in our setup could not be run to convergence, the algorithm converges only to the neighborhood of true solution, which is not stated in theory. At the same time, it is worth noting that a wide range of algorithm parameters were considered and some of its modifications were considered too in order to run it to the full convergence.

4.2 Linear regression in the vertical regime

In the case of vertical data partition, again, we present our target function as:

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) = \frac{1}{s} \sum_{i=1}^{s} l_i(a_i^T x, y_i) + \frac{\lambda}{2}\|x\|_2^2 \right\},$$

where $a_i \in \mathbb{R}^d, y_i \in \{-1, 1\}, \forall i \in 1, \ldots, s,\ \lambda = L/100.\ l_i : \mathbb{R}^2 \to \mathbb{R}$: $l_i(a_i^T x, y_i) = (a_i^T x - y_i)^2$.

We also conducted experiments to compare `DVPL-Katyusha` in combination with Rand1%, Natural Dithering Horváth et al. [2019] and PermK with vertical versions of regular gradient descent and its Nesterov acceleration in the same distributed setting. Number of workers in the experiments we conducted is $n = 5$. And again, on the x-axis, we postpone either $\frac{b}{\beta}$ or s for each iteration, without taking into account the coefficient $2\frac{n-1}{n}$, which is the same for all implementations.

Overall, it is clear from the graph that there is a significant acceleration of convergence compared to the naive implementation of the Nesterov approach in a vertically distributed mode (when devices share all s values of their local models, after which they calculate their part of the gradient and produce a Nesterov descent based on it) and with the same implementation of ordinary gradient descent.

In addition, the superiority of the PermK compressor is no longer present in this case, and the Rand1% compressor comes out on top, which emphasizes the different importance of compressors' properties for different data division modes.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

A. Alacaoglu and Y. Malitsky. Stochastic variance reduction for variational inequality methods, 2022.

D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding, 2017.

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018.

A. Beznosikov and A. Gasnikov. Similarity, compression and local steps: Three pillars of efficient communications for distributed variational inequalities, 2023.

A. Beznosikov, P. Richtárik, M. Diskin, M. Ryabinin, and A. Gasnikov. Distributed methods with compressed communication for solving variational inequalities, with theoretical guarantees, 2023.

A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan. On biased compression for distributed learning, 2024.

D. Cai, T. Fan, Y. Kang, L. Fan, M. Xu, S. Wang, and Q. Yang. Accelerating vertical federated learning. *IEEE Transactions on Big Data*, page 1–10, 2022. ISSN 2372-2096. doi: 10.1109/tbdata.2022.3192898. URL http://dx.doi.org/10.1109/TBDATA.2022.3192898.

T. Castiglia, A. Das, S. Wang, and S. Patterson. Compressed-vfl: Communication-efficient learning with vertically partitioned data, 2023.

E. Chan, M. Heimlich, A. Purkayastha, and R. Van De Geijn. Collective communication: theory, practice, and experience. *Concurrency and Computation: Practice and Experience*, 19(13): 1749–1783, 2007.

W. Chen, G. Ma, T. Fan, Y. Kang, Q. Xu, and Q. Yang. Secureboost+ : A high performance gradient boosting tree framework for large scale vertical federated learning, 2021.

T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX symposium on operating systems design and implementation (OSDI 14)*, pages 571–582, 2014.

E. Gorbunov. Distributed and stochastic optimization methods with gradient compression and local steps, 2021.

E. Gorbunov, K. Burlachenko, Z. Li, and P. Richtárik. Marina: Faster non-convex distributed learning with compression, 2022.

P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang. Privacy-preserving asynchronous federated learning algorithms for multi-party vertically collaborative learning, 2020.

Y. He, X. Huang, and K. Yuan. Unbiased compression saves communication in distributed optimization: When and how much? *Advances in Neural Information Processing Systems*, 36, 2024.

S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik. Stochastic distributed learning with gradient quantization and variance reduction, 2019.

L. Huang, Z. Li, J. Sun, and H. Zhao. Coresets for vertical federated learning: Regularized linear regression and k-means clustering, 2022.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Paper.pdf.

J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency, 2017.

D. Kovalev, S. Horvath, and P. Richtarik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop, 2019.

Z. Li, D. Kovalev, X. Qian, and P. Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. *arXiv preprint arXiv:2002.11364*, 2020.

Z. Li, H. Bao, X. Zhang, and P. Richtárik. Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization, 2021.

K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik. Distributed learning with compressed gradient differences, 2023.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi: 10.1137/100802001. URL https://doi.org/10.1137/100802001.

L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

P. Richtárik and M. Takáč. Distributed coordinate descent method for learning with big data, 2013.

V. Smith, S. Forte, C. Ma, M. Takáč, M. I. Jordan, and M. Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18(230):1–49, 2018.

J. Sun, Z. Xu, D. Yang, V. Nath, W. Li, C. Zhao, D. Xu, Y. Chen, and H. R. Roth. Communication-efficient vertical federated learning with limited overlapping samples, 2023.

R. Szlendak, A. Tyurin, and P. Richtárik. Permutation compressors for provably faster distributed nonconvex optimization, 2021.

J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2):1–33, 2020.

W. Xu, H. Fan, K. Li, and K. Yang. Efficient batch homomorphic encryption for vertically federated xgboost, 2021.

Q. Zhang, B. Gu, C. Deng, S. Gu, L. Bo, J. Pei, and H. Huang. Asysqn: Faster vertical federated learning algorithms with better computation resource utilization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3917–3927, 2021.

Supplementary Material

Contents

A Auxiliary Lemmas

Lemma 8. (Lemma 12 from [Kovalev et al., 2019]) For a random vector $x \in \mathbb{R}^d$ and any $y \in \mathbb{R}^d$, the variance of y can be represented in a form:

$$\mathbb{E}\left[\|x - \mathbb{E}\left[x\right]\|^2\right] = \mathbb{E}\left[\|x - \|y\|^2\right] - \mathbb{E}\left[\|\mathbb{E}\left[x\right] - y\|^2\right] \tag{6}$$

The next lemma is a form of Young's inequality:

Lemma 9. Let $\beta = \frac{\eta\theta_1}{L(1-\eta\theta_1)}$. Then $\forall a, b \in \mathbb{R}^d$ the following inequality holds:

$$\langle a, b\rangle \geq -\frac{\|a\|^2}{2\beta} - \frac{\beta\|b\|^2}{2} \tag{7}$$

B Horizontal partition

Although this might seem trivial, we further rely on gradient unbiasedness. Thus,

Lemma 10. In Algorithm 1 $\mathbb{E}\left[g^k\right] = \nabla f(x^k)$.

Proof: From Algorithm 1 we get:

$$g^k := \frac{1}{n}\sum_{i=1}^{n} Q_i\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right) + \nabla f(w^k) \tag{8}$$

Using Tower Property and Definition 1.1 of compressors' unbiasedness:

$$\mathbb{E}\left[g^k\right] := \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[Q_i\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right)\right] + \nabla f(w^k) = \nabla f(x^k)$$

Lemma 11. (Revised Lemma 6 from [Kovalev et al., 2019], proof for Lemma 2) In Algorithm 1 the following holds:

$$\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] \leq \frac{2L\omega}{n}\left(f(w^k) - f(x^k) - \langle\nabla f(x^k); w^k - x^k\rangle\right) \tag{9}$$

$$\tilde{L} = \frac{L\omega}{n}$$

Proof: From Algorithm 1, we derive 8, thus, with an empty term, after opening the square and taking advantage of the unbiased and independent compression operators in the scalar product:

$$\begin{aligned}
\mathbb{E}\left[\left\|g^k - \nabla f(x^k)\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} Q_i^w\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\} + \underline{\nabla f(w^k)} - \underline{\nabla f(x^k)}\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\left(Q_i\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\} - \left\{\underline{\nabla f_i(x^k)} - \underline{\nabla f_i(w^k)}\right\}\right)\right\|^2\right] \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\left\|Q_i\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\} - \left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\}\right\|^2\right]
\end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{1}{n^2}\sum_{i\neq l}\mathbb{E}[\langle Q_i\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}-\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}, Q_l\{\nabla f_l(x^k) \\
&- \quad \nabla f_l(w^k)\}-\left\{\nabla f_l(x^k)-\nabla f_l(w^k)\right\}\rangle] \\
&= \quad \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\left\|Q_i\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}-\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}\right\|^2\right]
\end{aligned}
$$

Next, using Assumption 1:

$$
\begin{aligned}
\mathbb{E}\left[\left\|g^k-\nabla f(x^k)\right\|^2\right] &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\left\|Q_i\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}-\left\{\nabla f_i(x^k)-\nabla f_i(w^k)\right\}\right\|^2\right] \\
&\leq \frac{\omega}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f_i(x^k)-\nabla f_i(w^k)\right\|^2\right]
\end{aligned}
$$

And finally, using Lipschitz property (Assumption 1) and convexity (Assumption 2) of functions $f_i(x)$:

$$
\begin{aligned}
\mathbb{E}\left[\left\|g^k-\nabla f(x^k)\right\|^2\right] &\leq \frac{\omega}{n^2}\sum_{i=1}^{n}2L\left(f_i(w^k)-f_i(x^k)-\langle\nabla f_i(x^k);w^k-x^k\rangle\right) \\
&\leq \frac{2L\omega}{n}\left(f(w^k)-f(x^k)-\langle\nabla f(x^k);w^k-x^k\rangle\right)
\end{aligned}
$$

□

Lemmas 7, 8, as already stated in the main section, are obtained from the original lemmas by replacing (only when $\frac{\omega}{n}>1$) the Lipschitz constant of the gradient with a constant $\frac{\omega}{n}$ times greater than the one given by assumption 1. We now need to show that nothing changes in how lemmas 7, 8 behave at their core, although an Algorithm has experienced slight changes:

Lemma 12. (Revised Lemma 7 from [Kovalev et al., 2019]) In Algorithm 1 the following holds:

$$
\langle g^k, x^*-z^{k+1}\rangle+\frac{\mu}{2}\left\|x^k-x^*\right\|^2\geq\frac{\widetilde{L}}{2\mu}\left\|z^k-z^{k+1}\right\|^2+Z^{k+1}-\frac{1}{1+\eta\sigma}Z^k \qquad (10)
$$

Proof: From Algorithm 1:

$$
z^{k+1}:=\frac{1}{1+\eta\frac{\mu}{\widetilde{L}}}\left(\eta\frac{\mu}{\widetilde{L}}x^k+z^k-\frac{\eta}{\widetilde{L}}g^k\right)
$$

$$
\to g^k=\frac{\widetilde{L}}{\mu}\left(z^k-z^{k+1}\right)+\mu\left(x^k-z^{k+1}\right)
$$

Thus (we will simply neglect the underlined summand):

$$
\begin{aligned}
\langle g^k, z^{k+1}-x^*\rangle &= \mu\langle x^k-z^{k+1}, z^{k+1}-x^*\rangle+\frac{\widetilde{L}}{\eta}\langle z^k-z^{k+1}, z^{k+1}-x^*\rangle \\
&= \frac{\mu}{2}\left(\|x^k-x^*\|^2-\underline{\|x^k-z^{k+1}\|^2}-\|z^{k+1}-x^*\|^2\right) \\
&+ \frac{\widetilde{L}}{2\eta}\left(\|z^k-x^*\|^2-\|z^k-z^{k+1}\|^2-\|z^{k+1}-x^*\|^2\right) \\
&\leq \frac{\mu}{2}\|x^k-x^*\|^2+\frac{\widetilde{L}}{2\eta}\left(\|z^k-x^*\|^2-(1+\eta\sigma)\|z^{k+1}-x^*\|^2\right)-\frac{\widetilde{L}}{2\eta}\|z^k-z^{k+1}\|^2
\end{aligned}
$$

□

Which concludes the proof

Now let us show that in the case $\frac{\omega}{n} > 1$ Lemma 8 can be rewritten in the same manner, replacing L - Lipschitz constant with the new $\widetilde{L}$.

Lemma 13. (Revised Lemma 8 from [Kovalev et al., 2019] We have:

$$\frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right) - \frac{\theta_2}{2\widetilde{L}\theta_1}\left\|g^k - \nabla f(x^k)\right\|^2 \leq \frac{\widetilde{L}}{2\eta}\left\|z^{k+1} - z^k\right\|^2 + \langle g^k, z^{k+1} - z^k\rangle \tag{11}$$

Proof: Firstly, we will prove this lemma in the case of $\frac{\omega}{n} > 1$. From Algorithm 1, one can see that $y^{k+1} \leftarrow x^k + \theta_1(z^{k+1} - z^k)$. Below we will introduce an empty term, use $\widetilde{L}$-smoothness of function $f(x)$ and, after that, utilize Lemma 7, where $\beta = \frac{\eta\theta_1}{\widetilde{L}(1-\eta\theta_1)}$:

$$\begin{aligned}
\frac{\widetilde{L}}{2\eta}\|z^{k+1} - z^k\|^2 + \langle g^k, z^{k+1} - z^k\rangle &= \frac{1}{\theta_1}\left(\frac{\widetilde{L}}{2\eta\theta_1}\|\theta_1\left(z^{k+1} - z^k\right)\|^2 + \langle g^k, \theta_1\left(z^{k+1} - z^k\right)\rangle\right) \\
&= \frac{1}{\theta_1}\left(\frac{\widetilde{L}}{2\eta\theta_1}\|y^{k+1} - x^k\|^2 + \langle g^k, y^{k+1} - x^k\rangle\right) \\
&= \underline{\frac{\widetilde{L}}{2\theta_1}\|y^{k+1} - x^k\|^2} + \frac{1}{\theta_1}\langle \underline{\nabla f(x^k)}, y^{k+1} - x^k\rangle \\
&+ \frac{\widetilde{L}}{2\theta_1}\left(\frac{1}{\eta\theta_1}\underline{-1}\right)\|y^{k+1} - x^k\|^2 + \frac{1}{\theta_1}\langle g^k \underline{-\nabla f(x^k)}, y^{k+1} - x^k\rangle) \\
&\geq \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right) + \frac{\widetilde{L}}{2\theta_1}\left(\frac{1}{\eta\theta_1} - 1\right)\|y^{k+1} - x^k\|^2 \\
&+ \frac{1}{\theta_1}\langle g^k - \nabla f(x^k), y^{k+1} - x^k\rangle \\
&\geq \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\eta\theta_1}{2\widetilde{L}(1-\eta\theta_1)}\|g^k - \nabla f(x^k)\|^2\right) \\
&= \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\theta_2}{2\widetilde{L}}\|g^k - \nabla f(x^k)\|^2\right)
\end{aligned}$$

The last equality was received due to $\eta = \frac{\theta_2}{(1+\theta_2)\theta_1}$. Thus, we derive:

$$\frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right) - \frac{\theta_2}{2\widetilde{L}\theta_1}\|g^k - \nabla f(x^k)\|^2 \leq \frac{\widetilde{L}}{2\eta}\|z^{k+1} - z^k\|^2 + \langle g^k, z^{k+1} - z^k\rangle$$

This is a part of the lemma for the case $\frac{w}{n} \leq 1$, e.g. $\widetilde{L} \leq L$.

$$\begin{aligned}
\frac{\widetilde{L}}{2\eta}\|z^{k+1} - z^k\|^2 + \langle g^k, z^{k+1} - z^k\rangle &= \frac{1}{\theta_1}\left(\frac{\widetilde{L}}{2\eta\theta_1}\|\theta_1\left(z^{k+1} - z^k\right)\|^2 + \langle g^k, \theta_1\left(z^{k+1} - z^k\right)\rangle\right) \\
&= \frac{1}{\theta_1}\left(\frac{\widetilde{L}}{2\eta\theta_1}\|y^{k+1} - x^k\|^2 + \langle g^k, y^{k+1} - x^k\rangle\right) \\
&= \frac{L}{2\theta_1}\|y^{k+1} - x^k\|^2 + \frac{1}{\theta_1}\langle \nabla f(x^k), y^{k+1} - x^k\rangle
\end{aligned}$$

$$
\begin{aligned}
+ \quad & \frac{L}{2\theta_1}\left(\frac{1}{\frac{L}{\widetilde{L}}\eta\theta_1} - 1\right)\|y^{k+1} - x^k\|^2 + \frac{1}{\theta_1}\langle g^k - \nabla f(x^k), y^{k+1} - x^k\rangle) \\
\geq \quad & \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right) + \frac{L}{2\theta_1}\left(\frac{1}{\frac{L}{\widetilde{L}}\eta\theta_1} - 1\right)\|y^{k+1} - x^k\|^2 \\
+ \quad & \frac{1}{\theta_1}\langle g^k - \nabla f(x^k), y^{k+1} - x^k\rangle \\
\geq \quad & \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\frac{L}{\widetilde{L}}\eta\theta_1}{2\widetilde{L}(1 - \frac{L}{\widetilde{L}}\eta\theta_1)}\|g^k - \nabla f(x^k)\|^2\right) \\
= \quad & \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\theta_2}{2L}\|g^k - \nabla f(x^k)\|^2\right)
\end{aligned}
$$

The last inequality was obtained by using Young's inequality in the form of $\langle a, b\rangle \geq -\frac{\|a\|^2}{2\beta} - \frac{\beta\|b\|^2}{2}$ for $\beta = \frac{\frac{L}{\widetilde{L}}\eta\theta_1}{L\left(1 - \frac{L}{\widetilde{L}}\eta\theta_1\right)}$.

The last equality was received due to $\eta = \frac{\frac{\widetilde{L}}{L}\theta_2}{(1+\theta_2)\theta_1}$.
Using that $\widetilde{L} \leq L$, we get that:

$$
\frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\theta_2}{2L}\|g^k - \nabla f(x^k)\|^2\right) \geq \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k) - \frac{\theta_2}{2\widetilde{L}}\|g^k - \nabla f(x^k)\|^2\right)
$$

And finally we get:

$$
\frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right) - \frac{\theta_2}{2\widetilde{L}\theta_1}\|g^k - \nabla f(x^k)\|^2 \leq \frac{\widetilde{L}}{2\eta}\|z^{k+1} - z^k\|^2 + \langle g^k, z^{k+1} - z^k\rangle
$$

□

Lemma 9, however, does not require any changes in the proof at all, because it is based only on the definition of w^{k+1} from Algorithm 1.

Lemma 14. (Lemma 9 from [Kovalev et al., 2019]) We have:

$$
\mathbb{E}\left[f(w^{k+1})\right] = (1-p)\,f(w^k) + pf(y^k) \tag{12}
$$

Lemma 15. (Revised Lemma 10 from [Kovalev et al., 2019], proof for Theorem 1) Considering lemmas 11–14, we get:

$$
\mathbb{Z}^k\left[\frac{1}{1+\eta\sigma}\right] + \mathbb{Y}^k\left[(1 - \theta_1(1-\theta_2))\right] + \mathbb{W}^k\left[1 - \frac{p\theta_1}{1+\theta_1}\right] \geq \mathbb{E}\left[\mathbb{Z}^{k+1} + \mathbb{Y}^{k+1} + \mathbb{W}^{k+1}\right] \tag{13}
$$

Proof: Using strong convexity (Assumption 2) and adding an empty term, we get:

$$
\begin{aligned}
f(x^*) \quad & \geq \quad f(x^k) + \langle\nabla f(x^k), x^* - x^k\rangle + \frac{\mu}{2}\|x^k - x^*\|^2 \\
& \geq \quad f(x^k) + \langle\nabla f(x^k), x^*\underline{-z^k} + \underline{z^k} - x^k\rangle + \frac{\mu}{2}\|x^k - x^*\|^2
\end{aligned}
$$

From Algorithm 1, $z^k - x^k := \frac{\theta_2}{\theta_1}\left(x^k - w^k\right) + \frac{1-\theta_1-\theta_2}{\theta_1}\left(x^k - y^k\right)$, thus:

$$
f(x^*) \quad \geq \quad f(x^k) + \langle\nabla f(x^k), x^*\underline{-z^k} + \underline{z^k} - x^k\rangle + \frac{\mu}{2}\|x^k - x^*\|^2
$$

$$
\begin{aligned}
&\geq\ f(x^k) + \frac{\mu}{2}\|x^k - x^*\|^2 + \langle \nabla f(x^k), x^* - z^k\rangle \\
&+\ \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \underline{\frac{1-\theta_1-\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - y^k\rangle}
\end{aligned}
$$

Applying convexity to the underlined term, utilizing unbiasedness of the gradient and adding an empty term, we derive:

$$
\begin{aligned}
f(x^*) \ &\geq\ f(x^k) + \frac{\mu}{2}\|x^k - x^*\|^2 + \langle \nabla f(x^k), x^* - z^k\rangle \\
&+\ \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \underline{\frac{1-\theta_1-\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - y^k\rangle} \\
&\geq\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[\frac{\mu}{2}\|x^k - x^*\|^2 \langle g^k, x^* \underline{- z^{k+1}}\rangle + \langle g^k, \underline{z^{k+1}} - z^k\rangle\right] \\
&=\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[\frac{\mu}{2}\|x^k - x^*\|^2 + \langle g^k, x^* - z^{k+1}\rangle + \langle , z^{k+1} - z^k\rangle\right]
\end{aligned}
$$

Using Lemma 12, we get:

$$
\begin{aligned}
f(x^*) \ &=\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[\frac{\mu}{2}\|x^k - x^*\|^2 + \langle g^k, x^* - z^{k+1}\rangle + \langle g^k, z^{k+1} - z^k\rangle\right] \\
&\geq\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k\right] + \mathbb{E}\left[\langle g^k, z^{k+1} - z^k\rangle + \frac{\widetilde{L}}{2\eta}\|z^k - z^{k+1}\|^2\right] \\
&=\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k\right] + \underline{\mathbb{E}\left[\langle g^k, z^{k+1} - z^k\rangle + \frac{\widetilde{L}}{2\eta}\|z^k - z^{k+1}\|^2\right]}
\end{aligned}
$$

Utilizing Lemma 13, we transform an underlined summand:

$$
\begin{aligned}
f(x^*) \ &=\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k\right] + \underline{\mathbb{E}\left[\langle g^k, z^{k+1} - z^k\rangle + \frac{\widetilde{L}}{2\eta}\|z^k - z^{k+1}\|^2\right]} \\
&\geq\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k\right] + \underline{\mathbb{E}\left[\frac{1}{\theta_1}\Big(f(y^{k+1}) - f(x^k)\Big) - \frac{\theta_2}{2\widetilde{L}\theta_1}\|g^k - \nabla f(x^k)\|^2\right]} \\
&\geq\ f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\Big(f(x^k) - f(y^k)\Big) \\
&+\ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k + \frac{1}{\theta_1}\Big(f(y^{k+1}) - f(x^k)\Big)\right] - \mathbb{E}\left[\frac{\theta_2}{2\widetilde{L}\theta_1}\|g^k - \nabla f(x^k)\|^2\right]
\end{aligned}
$$

Finally, using Lemma 11 to evaluate the last summ and, we get:

$$\begin{aligned}
f(x^*) &\geq f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\left(f(x^k) - f(y^k)\right) \\
&+ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k + \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right)\right] - \mathbb{E}\left[\frac{\theta_2}{2\widetilde{L}\theta_1}\|g^k - \nabla f(x^k)\|^2\right] \\
&\geq f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\left(f(x^k) - f(y^k)\right) \\
&+ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k + \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right)\right] \\
&- \frac{\theta_2}{2\widetilde{L}\theta_1}\underline{\mathbb{E}\Big[2\widetilde{L}\left(f(w^k) - f(x^k) - \langle \nabla f(x^k); w^k - x^k\rangle\right)\Big]} \\
&= f(x^k) + \frac{\theta_2}{\theta_1}\langle \nabla f(x^k), x^k - w^k\rangle + \frac{1-\theta_1-\theta_2}{\theta_1}\left(f(x^k) - f(y^k)\right) \\
&+ \mathbb{E}\left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k + \frac{1}{\theta_1}\left(f(y^{k+1}) - f(x^k)\right)\right] \\
&- \frac{\theta_2}{\theta_1}\mathbb{E}\left[f(w^k) - f(x^k) - \langle \nabla f(x^k); w^k - x^k\rangle\right] \\
&= -\frac{\theta_2}{\theta_1}f(w^k) - \frac{1-\theta_1-\theta_2}{\theta_1}f(y^k) + \left[Z^{k+1} - \frac{1}{1+\eta\sigma}Z^k + \frac{1}{\theta_1}f(y^{k+1})\right]
\end{aligned}$$

As one can see, this result corresponds exactly to the formulation of Lemma 10 from the original article, with the only difference that now everywhere we have the constant $\widetilde{L}$ instead of L, which leads us to the same final expression:

$$\mathbb{Z}^k\left[\frac{1}{1+\eta\sigma}\right] + \mathbb{Y}^k\left[(1-\theta_1(1-\theta_2))\right] + \mathbb{W}^k\left[1 - \frac{p\theta_1}{1+\theta_1}\right] \geq \mathbb{E}\left[\mathbb{Z}^{k+1} + \mathbb{Y}^{k+1} + \mathbb{W}^{k+1}\right] \tag{14}$$

□

From Lemma 15 we get our main Theorem directly by substituting into the inequality the specified values that preserve the inequality:

Theorem 4. (Theorem 11 from [Kovalev et al., 2019]) Let Assumptions 1, 2 be hold. Additionally, let $p = \frac{1}{n}$, $\theta_1 = \min\{\sqrt{\frac{2\sigma n}{3}}, \frac{1}{2}\}$, $\theta_2 = \frac{1}{2}$. Then $\mathbb{E}\left[\Psi^k\right] \leq \epsilon\Psi^0$ after $K = \mathcal{O}\left(\left(n + \sqrt{\frac{\omega L}{\mu}}\right)\log\frac{1}{\epsilon}\right)$ iterations.

Theorem 5. (The efficient Lipschitz for PermK compressor in the horizontal case, proof for Lemma 4)
For `DHPL-Katyusha` with PermK compressor, the efficient Lipschitz constant is equal to L.

Proof: Denote Q_i as a PermK compressor that operates on i-th function. Therefore, we get:

$$\begin{aligned}
\mathbb{E}\left[\left\|g^k - \nabla f(x^k)\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} Q_i^w\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\} + \nabla f(w^k) - \nabla f(x^k)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} Q_i\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\} - \frac{1}{n}\sum_{i=1}^{n}\left\{\nabla f_i(x^k) - \nabla f_i(w^k)\right\}\right\|^2\right]
\end{aligned}$$

$$
\begin{aligned}
&\le \mathbb{E}\left[A\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(x^k)-\nabla f_i(w^k)\right\|^2 - B\left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x^k)-\nabla f_i(w^k)\right\|^2\right] \\
&\le \mathbb{E}\left[A\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(x^k)-\nabla f_i(w^k)\right\|^2\right] \\
&\le 2LA\left(f(w^k)-f(x^k)-\langle\nabla f(x^k), w^k-x^k\rangle\right)
\end{aligned}
$$

The first inequality is obtained by using A-B inequality from [Szlendak et al., 2021], the second inequality we get by utilizing the positive definiteness of the norm and the last inequality is obtained by using an analogy with 10.

In the case $n \geq d$ we have $A = \frac{d-1}{n-1} \leq 1$, and in the case $n < d$ we have $A = 1$. Combining these cases, we get the estimation for the efficient Lipschitz constant as 1.

C Vertical case

In this section, we provide the complete proofs for our results in the vertical case.

Lemma 16. (Revised Lemma 6 from [Kovalev et al., 2019]) In Algorithm 2 the following holds:

$$\begin{aligned}&\mathbb{E}\left\|g^k(x^k) - \nabla\mathcal{L}\left(Ax^k, b_j\right)\right\|^2 \\ &\leq 2\max\left\{L, \tfrac{\bar{L}}{K}\right\}\left(\nabla\mathcal{L}\left(Aw^k, b\right) - \nabla\mathcal{L}\left(Ax^k, b\right) - \left\langle\nabla\mathcal{L}\left(Ax^k, b\right); w^k - x^k\right\rangle\right)\end{aligned}$$

Proof:

$$\begin{aligned}
\mathbb{E}||g^k(x^k) \quad &- \quad \nabla\mathcal{L}\left(Ax^k, b_j\right)||^2 \\
&= \mathbb{E}_{J^k}\left\|\frac{1}{K}\sum_{j\in J^k}\frac{1}{sp_j}\left[\nabla\mathcal{L}_j\left(\sum_{i=1}^n X_{ij}, b_j\right) - \nabla\mathcal{L}_j\left(\sum_{i=1}^n W_{ij}, b_j\right)\right] + \nabla\mathcal{L}\left(Aw^k, b_j\right) - \nabla\mathcal{L}\left(Ax^k, b_j\right)\right\|^2 \\
&= \frac{1}{K}\mathbb{E}_{i\sim D}\left\|\left(\frac{1}{sp_j}\left[\nabla\mathcal{L}_j\left(\sum_{i=1}^n X_{ij}, b_j\right) - \nabla\mathcal{L}_j\left(\sum_{i=1}^n W_{ij}, b_j\right)\right]\right) - \left(\nabla\mathcal{L}\left(Aw^k, b_j\right) + \nabla\mathcal{L}\left(Ax^k, b_j\right)\right)\right\|^2 \\
&\leq \frac{1}{K}\mathbb{E}_{i\sim D}\left\|\frac{1}{sp_j}\left[\nabla\mathcal{L}_j\left(\sum_{i=1}^n X_{ij}, b_j\right) - \nabla\mathcal{L}_j\left(\sum_{i=1}^n W_{ij}, b_j\right)\right]\right\|^2 \\
&\leq \frac{1}{K}\sum_{j=1}^s\frac{2L_j}{s^2p_j}\left(\nabla\mathcal{L}\left(Aw^k, b\right) - \nabla\mathcal{L}\left(Ax^k, b\right) - \left\langle\nabla\mathcal{L}\left(Ax^k, b\right); w^k - x^k\right\rangle\right) \\
&= \frac{2\bar{L}}{K}\left(\nabla\mathcal{L}\left(Aw^k, b\right) - \nabla\mathcal{L}\left(Ax^k, b\right) - \left\langle\nabla\mathcal{L}\left(Ax^k, b\right); w^k - x^k\right\rangle\right) \\
&\leq 2\max\left\{L, \frac{\bar{L}}{K}\right\}\left(\nabla\mathcal{L}\left(Aw^k, b\right) - \nabla\mathcal{L}\left(Ax^k, b\right) - \left\langle\nabla\mathcal{L}\left(Ax^k, b\right); w^k - x^k\right\rangle\right)
\end{aligned}$$

Lemma 17. (Revised Lemma 6 from [Kovalev et al., 2019]) In Algorithm 3 the following holds:

$$\begin{aligned}\mathbb{E}||g^k(x^k) - \nabla f(x^k)||^2 \quad &\leq \quad 2L\left(f(w^k) - f(x^k) - \langle\nabla f(x^k); w^k - x^k\rangle\right) \\ &\cdot \quad \left(1 + (\omega - 1)\frac{s\sum\limits_{j=1}^s L_j^2}{\mu^2}\right)\end{aligned}$$

Proof:

$$\begin{aligned}
\mathbb{E}\left[\left\|g^k(x^k) - \nabla f(x^k)\right\|^2\right] \quad &= \quad \mathbb{E}\left[\left\|\frac{1}{b_s}\sum_{j\in\{J\}}\left[2\sum_{i=1}^n Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right]A_j^T + \nabla f(w^k) - \nabla f(x^k)\right\|^2\right] \\
&\leq \quad \frac{1}{b_s^2}\cdot b_s\sum_{j\in J}\cdot\mathbb{E}\left[\left\|2\sum_{i=1}^n Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)A_j^T + \nabla f(w^k) - \nabla f(x^k)\right\|^2\right]
\end{aligned}$$

Here we used a Cauchy-Bunyakovsky-Schwarz inequality. The next step is to evaluate the term within mathematic expectancy.

$$
\begin{aligned}
& \mathbb{E}\left[\left\|2\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right) A_j^T + \nabla f(w^k) - \nabla f(x^k)\right\|^2\right] \\
= \; & \mathbb{E}\left[\left\|2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right) - \frac{2}{s}A^T A(x^k - w^k)\right\|^2\right] \\
= \; & \mathbb{E}\left[\left\|2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right\|^2\right] + \mathbb{E}\left[\left\|\frac{2}{s}A^T A(x^k - w^k)\right\|^2\right] \\
- \; & 2\cdot\mathbb{E}\left[\langle 2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right), \frac{2}{s}A^T A(x^k - w^k)\rangle\right] \\
= \; & \mathbb{E}\left[\left\|2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right\|^2\right] - \left\|\frac{2}{s}A^T A(x^k - w^k)\right\|^2 \\
\leq \; & \mathbb{E}\left[\left\|2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right\|^2\right]
\end{aligned}
$$

Now, using norm properties and Assumption 1.1 we get:

$$
\begin{aligned}
& \mathbb{E}\left[\left\|2A_j^T\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right\|^2\right] \\
= \; & 4\mathbb{E}\left[\|A_j^T\|^2\left|\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right|^2\right] \\
= \; & 4\mathbb{E}\left[\|A_j^T\|^2\left|\sum_{i=1}^{n}\left(Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\right)^2 + \sum_{i\neq t} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right) Q_t\left(\langle A_{jt}^T, x_t^k - w_t^k\rangle\right)\right|\right] \\
\leq \; & 4\mathbb{E}\left[\|A_j^T\|^2\left|\sum_{i=1}^{n}\omega(\langle A_{ji}^T, x_i^k - w_i^k\rangle)^2 + \sum_{i\neq t}\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)\left(\langle A_{jt}^T, x_t^k - w_t^k\rangle\right)\right|\right] \\
= \; & 4\mathbb{E}\left[\|A_j^T\|^2\left|\sum_{i=1}^{n}(\omega - 1)(\langle A_{ji}^T, x_i^k - w_i^k\rangle)^2 + \left(\langle A_j^T, x^k - w^k\rangle\right)^2\right|\right] \\
= \; & \mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + 4(\omega - 1)\|A_j^T\|^2\sum_{i=1}^{n}\left|\langle A_{ji}^T, x_i^k - w_i^k\rangle\right|^2\right] \\
\leq \; & \mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + 4(\omega - 1)\|A_j^T\|^4\left\|x^k - w^k\right\|^2\right] \\
= \; & \mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + 4(\omega - 1)\|A_j^T\|^4\left\|\left(\frac{2}{s}A^T A\right)^{-1}(\nabla f(x^k) - \nabla f(w^k))\right\|^2\right] \\
\leq \; & \mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + (\omega - 1)\|A_j^T\|^4 s^2\left\|\left(A^T A\right)^{-1}\right\|^2\left\|\nabla f(x^k) - \nabla f(w^k)\right\|^2\right] \\
\leq \; & \mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + (\omega - 1)\|A_j^T\|^4\frac{s^2}{\mu^2}\left\|\nabla f(x^k) - \nabla f(w^k)\right\|^2\right]
\end{aligned}
$$

$$\leq \left(f(w^k) - f(x^k) - \langle \nabla f(x^k); w^k - x^k\rangle\right)\left(2L + 2L(\omega-1)\frac{s^2 \cdot \frac{\sum\limits_{j=1}^{s} L_j^2}{s}}{\mu^2}\right)$$

$$= 2L\left(f(w^k) - f(x^k) - \langle \nabla f(x^k); w^k - x^k\rangle\right)\left(1 + (\omega-1)\frac{s \cdot \sum\limits_{j=1}^{s} L_j^2}{\mu^2}\right)$$

Where the first equality is obtained by using absolute homogeneity of vector norm and having that $\sum_{i=1}^{n} Q_i\left(\langle A_{ji}^T, x_i^k - w_i^k\rangle\right)$ is a scalar value, the first inequality is obtained by using compressor property 1.1, the second and the third inequalities are obtained by using the definition of the norm as the supremum, the fourth inequality is obtained by using that the spectral norm of matrix is bounded by the biggest eigenvalue, which is bounded by $\frac{1}{\mu}$, final inequality is obtained by using the Lipschitz property 1.

Lemma 18. There exists such function that holds under 5 that its efficient Lipschitz constant $\widetilde{L}$ is proportional to $(\omega-1)\frac{L^3}{\mu^2}$.

Proof:
Assume the size of batch $b = 1$. Therefore, in previous lemma we have an exact quality:

$$\mathbb{E}\left[\|g^k - \nabla f(x^k)\|^2\right] = 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + (\omega-1)\left\|A_j^T\right\|^2 \sum_{i=1}^{n}\left|\langle a_j^i, x_i^k - w_i^k\rangle\right|^2\right]$$

Now assume matrix A is given as $(b < a)$:

$$A = \begin{pmatrix} a & a \\ b & -b \end{pmatrix}$$

Assume that $k = 2$ and on the first iteration of `DVPL-Katyusha` ∇f_2 was chosen and b (from $Ax - b$) and x_0 are collinear with A_2^T.
Denote $x^k - w^k = (c, -c)^T$. It has its first coordinate equal to the negative second, because it is orthogonal to A_1^T.

As A_1^T is orthogonal with $x^k - w^k$:

$$\nabla f_1(x^k) - \nabla f_1(w^k) = A_1^T\langle A_1^T, x^k - w^k\rangle = 0$$

Therefore, we have the difference of the target gradient in x^k and in w^k to be equal to:

$$\nabla f(x^k) - \nabla f(w^k) = \nabla f_1(x^k) - \nabla f_1(w^k) + \nabla f_2(x^k) - \nabla f_2(w^k) = \nabla f_2(x^k) - \nabla f_2(w^k)$$

Note that the norm of $w^k - x^k$ equals to:

$$||w^k - x^k||^2 = 2c^2 = \frac{||\nabla f_2(x^k) - \nabla f_2(w^k)||^2}{4||A_2^T||^4} = \frac{||\nabla f(x^k) - \nabla f(w^k)||^2}{4||A_2^T||^4}$$

Also note that eigenvalues of A^TA are equal to $2a^2$ and $2b^2$. Therefore $L = 2a^2$ and $\mu = 2b^2$. Combining all of these facts we get:

$$\begin{aligned}
& \mathbb{E}\left[||g^k - \nabla f(x^k)||^2\right] \\
= \; & 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2 + (\omega - 1)\left\|A_j^T\right\|^2 \sum_{i=1}^n \left|\langle a_j^i, x_i^k - w_i^k\rangle\right|^2\right] \\
= \; & 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2\right] + (\omega - 1)\left\|A_1^T\right\|^2\left(a^2c^2 + a^2c^2\right) + (\omega - 1)\left\|A_2^T\right\|^2\left(b^2c^2 + b^2c^2\right) \\
= \; & 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2\right] + 2(\omega - 1)\left\|A_1^T\right\|^4 c^2 + 2(\omega - 1)\left\|A_2^T\right\|^4 c^2 \\
= \; & 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2\right] + 2(\omega - 1)\left\|A_1^T\right\|^4 c^2 + 2(\omega - 1)\left\|A_2^T\right\|^4 c^2 \\
= \; & 4\mathbb{E}\left[\left\|\nabla f_j(x^k) - \nabla f_j(w^k)\right\|^2\right] + (\omega - 1)\frac{L^2}{2\mu^2}||\nabla f(x^k) - \nabla f(w^k)||^2 + \frac{1}{2}(\omega - 1)||\nabla f(x^k) - \nabla f(w^k)||^2
\end{aligned}$$

Using the Lipschitz property in analogue with 17, we conclude the proof.

C.1 PermK compressor

In PermK compressor, each worker chooses its own sample and will AllReduce its components only from it.

$$\begin{aligned}
\mathbb{E}\left\|g^k - \nabla f(x^k)\right\|^2 \; = \; & \mathbb{E}\left\|2\frac{n}{n}\sum_{i=1}^n A_i^T \sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij} - \frac{2}{s}A^TA\left(x^k - w^k\right)\right\|^2 \\
= \; & 4\mathbb{E}\left\|\sum_{i=1}^n A_i^T \sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij} - \frac{1}{s}A^TA\left(x^k - w^k\right)\right\|^2 \\
= \; & 4\mathbb{E}\left\|\sum_{i=1}^n \left[A_i^T \sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij} - \frac{1}{ns}A^TA\left(x^k - w^k\right)\right]\right\|^2 \\
\leq \; & 8\sum_{i=1}^n \mathbb{E}\left\|A_i^T \sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij} - \frac{1}{ns}A^TA\left(x^k - w^k\right)\right\|^2 \\
= \; & 8\sum_{i=1}^n \mathbb{E}\left[\left\|A_i^T \sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij}\right\|^2 - \left\|\frac{1}{ns}A^TA\left(x^k - w^k\right)\right\|^2\right] \\
\leq \; & 8\sum_{i=1}^n \mathbb{E}\left[\left(\sum_{j=1}^n \langle A_{ji}^T, x_{ji}^k - w_{ji}^k\rangle I_{ij}\right)^2 \left\|A_i^T\right\|^2\right]
\end{aligned}$$

$$= \quad 8\sum_{i=1}^{n}\mathbb{E}\left[\left(\sum_{j=1}^{n}(\langle A_{ji}^T,x_{ji}^k-w_{ji}^k\rangle I_{ij})^2+\sum_{j\neq t}(\langle A_{ji}^T,x_{ji}^k-w_{ji}^k\rangle I_{ij})(\langle A_{jt}^T,x_{ti}^k-w_{ti}^k\rangle I_{it})\right)\cdot\left\|A_i^T\right\|^2\right]$$

Using that $\mathbb{E}I_{ij}^2 = \frac{1}{n}$ and as any sample can be chosen only by one worker, we get that $\mathbb{E}I_{it}I_{ij} = 0, j \neq t$. Therefore, using the analogy with 17 we get:

$$\begin{aligned} &= \quad 8\sum_{i=1}^{n}\mathbb{E}\left[\frac{1}{n}\sum_{j=1}^{n}\langle A_{ji}^T,x_{ji}^k-w_{ji}^k\rangle^2\left\|A_i^T\right\|^2\right] \\ &\leq \quad 4L\left(f(w^k)-f(x^k)-\langle\nabla f(x^k);w^k-x^k\rangle\right)\frac{s\sum\limits_{j=1}^{s}L_j^2}{\mu^2} \end{aligned}$$