A Linearly Convergent Algorithm for Decentralized Optimization: Sending Less Bits for Free!

Dmitry Kovalev[1] Anastasia Koloskova[2] Martin Jaggi[2] Peter Richtarik [1] Sebastian U. Stich[2]

[1]KAUST [2]EPFL

Decentralized Minimization Problem

SETUP: $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ is an undirected connected network, where

- $\mathcal{V} := \{1, \dots, n\}$ is a set of computing nodes,
- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of communication links.

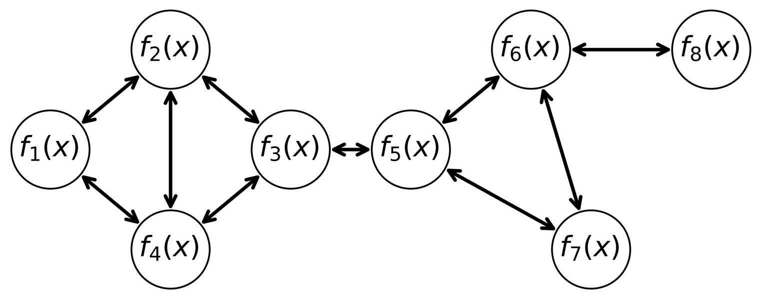

Figure 1: Example of the network $\mathcal{G}$ with $n = 8$ nodes

Each node $i \in \mathcal{V}$ owns function $f_i \colon \mathbb{R}^d \to \mathbb{R}$, which is L-smooth and μ-strongly convex. Let $\kappa = \frac{L}{\mu}$ be the condition number.

GOAL: Find solution of the minimization problem

$$x^* = \arg\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i \in \mathcal{V}} f_i(x). \tag{1}$$

Each node $i \in \mathcal{V}$ is allowed to calculate one of the gradient oracles and communicate $\mathcal{O}(1)$ compressed vectors of size d with each neighbor along the links $e \in \mathcal{E}$.

Gradient Oracles

Option A (dual gradient): We use dual gradient oracle $\nabla f_i^*(z)$, where $f_i^*(z)$ is the Fenchel transform of function $f(x)$. It is used when $\nabla f_i^*(z)$ can be computed efficiently.

Option B (primal gradient): We use primal gradient oracle $\nabla f_i(x)$.

Option C (primal stochastic gradient): When each function $f_i(x)$ is given as an expectation $\mathbb{E}_{\xi \sim \mathcal{D}}[f_i(x;\xi)]$, we use stochastic gradient oracle $\nabla f_i(x;\xi)$, where ξ is sampled from the distribution $\mathcal{D}$.

Option D (primal incremental gradient): When each function $f_i(x)$ is given as a finite sum $f_i(x) = \frac{1}{m}\sum_{j=1}^m f_{ij}(x)$, we use incremental gradient oracle $\nabla f_{ij}(x)$, where $j \in \{1, \dots, m\}$.

Compressed Communication

Communication is a key bottleneck in distributed training. We tackle it by forcing each node to apply compression operator $\mathcal{Q}$ to the vector $g \in \mathbb{R}^d$ it wants to send to a neighbour.

Compression Operator (ω-quantization)

A random operator $\mathcal{Q} \colon \mathbb{R}^d \to \mathbb{R}^d$ is called ω-quantization for $\omega \geq 0$, if it satisfies the following properties for all $g \in \mathbb{R}^d$:

$$\mathbb{E}\left[\mathcal{Q}(g)\right] = g, \qquad \mathbb{E}\left[\|\mathcal{Q}(g) - g\|^2\right] \leq \omega \|g\|^2.$$

Problem Reformulation

Problem (1) has an equivalent reformulation

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{W}\mathbf{X} = 0} F(\mathbf{X}), \tag{2}$$

where the function $F \colon \mathbb{R}^{n\times d} \to \mathbb{R}$ is defined by

$$F(\mathbf{X}) := \sum_{i \in \mathcal{V}} f_i(x_i), \qquad x_i \text{ is the } i\text{-th row of } \mathbf{X},$$

matrix $\mathbf{W} \in \mathbb{S}_+^n$ is a weighted Laplacian:

$$\mathbf{W}_{ij} = \begin{cases} 0, & i \neq j, (i,j) \notin \mathcal{E} \\ -w_{ij}, & i \neq j, (i,j) \in \mathcal{E} \\ \sum_{l \in \mathcal{N}_i} w_{il}, & i = j \end{cases},$$

where $\mathcal{N}_i := \{j \in \mathcal{V} \mid j \neq i, (i,j) \in \mathcal{E}\}$, $w_{ij} > 0$ for all $(i,j) \in \mathcal{E}$, $\mathbb{S}_+^n$ is a set of symmetric positive definite $n \times n$ matrices. Note that

$$\mathbf{W}\mathbf{X} = 0 \Leftrightarrow x_1 = \ldots = x_n,$$

which implies the equivalence of the problems (1) and (2). Problem (2) has an equivalent saddle-point reformulation:

$$\min_{\mathbf{X} \in \mathbb{R}^{n\times d}} \max_{\mathbf{Z} \in \mathcal{L}} \Lambda(\mathbf{X}, \mathbf{Z}) := F(\mathbf{X}) - \langle \mathbf{X}, \mathbf{Z} \rangle, \tag{3}$$

where $\mathcal{L} = \{\mathbf{Z} \in \mathbb{R}^{n\times d} \mid \sum_{j=1}^n z_j = 0, z_i \text{ is the } i\text{-th row of } \mathbf{Z}\}$, $\langle \mathbf{X}, \mathbf{Z} \rangle := \operatorname{trace}(\mathbf{X}\mathbf{Z})$ is a scalar product of $\mathbf{X}$ and $\mathbf{Z}$.

Main Algorithm

Algorithm 1

1: **Input:** $\mathbf{X}^0 \in \mathbb{R}^{n \times d}, \mathbf{Z}^0 \in \mathcal{L}, h_1^0, \dots, h_n^0 \in \mathbb{R}^d$ $\alpha, \eta, \theta > 0$.
2: **for** $k = 0, 1, 2, \dots$ **do**
3: Compute $\mathbf{X}^{k+1}$ ▷ Primal Step
4: **for** $i = 1, \dots, n$ **do** in parallel
5: **for** $j \in \mathcal{N}_i$ **do**
6: $\Delta_{ij} = \mathcal{Q}(x_i^{k+1} - h_i^k) + h_i^k$ ▷ Compression
7: **end for**
8: $h_i^{k+1} = h_i^k + \alpha \mathcal{Q}(x_i^{k+1} - h_i^k)$ ▷ Compression
9: $z_i^{k+1} = z_i^k - \theta \sum_{j \in \mathcal{N}_i} w_{ij}(\Delta_{ij}^k - \Delta_{ji}^k)$ ▷ Dual Step
10: **end for**
11: **end for**

References:

[1] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020.

[2] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[3] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[4] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing*, 67(19):4934–4947, 2019.

[5] Sulaiman A Alghunaim and Ali H Sayed. Linear convergence of primal–dual gradient methods and their performance in distributed optimization. *Automatica*, 117:109003, 2020.

[6] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487. PMLR, 2019.

Algorithm Description

Primal Step. This is a minimization step over $\mathbf{X}$ in the Problem (3). There are 4 options:

Option A performs the exact minimization:

$$\mathbf{X}^{k+1} = \arg\min_{\mathbf{X} \in \mathbb{R}^{n\times d}} \Lambda(\mathbf{X}, \mathbf{Z}^k) = \nabla F^*(\mathbf{Z}^k)$$

Option B performs inexact minimization with a single gradient step:

$$\begin{aligned} \mathbf{X}^{k+1} &= \mathbf{X}^k - \eta \nabla_{\mathbf{X}} \Lambda(\mathbf{X}^k, \mathbf{Z}^k) \\ &= \mathbf{X}^k - \eta(\nabla F(\mathbf{X}^k) - \mathbf{Z}^k). \end{aligned}$$

Option C performs inexact minimization with a stochastic gradient step:

$$x_i^{k+1} = x_i^k - \eta(\nabla f_i(x_i^k; \xi_i^k) - z_i^k) \text{ for each } i \in \mathcal{V},$$

where $\xi_i^k \sim \mathcal{D}$. By σ^2 we denote the stochastic gradient variance at the optimum:

$$\sigma^2 = \frac{1}{n} \sum_{i \in \mathcal{V}} \mathbb{E}_{\xi \sim \mathcal{D}}\left[\|\nabla f_i(x^*, \xi) - \nabla f_i(x^*)\|^2\right].$$

Option D performs inexact minimization with a variance reduced gradient step [1]:

$$x_i^{k+1} = x_i^k - \eta(\nabla f_{ij_i^k}(x_i^k) - \nabla f_{ij_i^k}(w_i^k) + \nabla f_i(w_i^k) - z_i^k),$$

$$w_i^{k+1} = \begin{cases} x_i^k, & \text{with probability } \frac{1}{m} \\ w_i^k, & \text{with probability } 1 - \frac{1}{m} \end{cases},$$

where j_i^k is sampled from $\{1, \dots, m\}$ uniformly at random.

Compression. We use compression step with variance reduction. It was originally used for centralized distributed training algorithms DIANA [2] and VR-DIANA [3].

Dual Step. This is a decentralized communication step with compression. It can be seen as a compressed version of a gradient ascent step in $\mathbf{Z}$ under metric $\|\cdot\|^2_{\mathbf{W}^\dagger}$ for problem (3). In particular,

$$\begin{aligned} \mathbb{E}_{\mathcal{Q}}\left[\mathbf{Z}^{k+1}\right] &= \mathbf{Z}^k + \theta \mathbf{W} \nabla_{\mathbf{Z}} \Lambda(\mathbf{X}^{k+1}, \mathbf{Z}^k) \\ &= \mathbf{Z}^k - \theta \mathbf{W}\mathbf{X}^{k+1}. \end{aligned}$$

Variance Bound (Key Lemma)

Let Σ^k be the variance of $\mathbf{Z}^{k+1}$:

$$\Sigma^k := \mathbb{E}_{\mathcal{Q}}\left[\|\mathbf{Z}^{k+1} - \mathbb{E}_{\mathcal{Q}}\left[\mathbf{Z}^{k+1}\right]\|^2_{\mathbf{W}^\dagger}\right].$$

Then the following inequality holds:

$$\Sigma^k \leq 4\theta^2 \omega \lambda_{\max}(\mathbf{W}) \rho_\infty \rho^{-1} \left[\|\mathbf{X}^{k+1} - \mathbf{X}^*\|^2 + \sum_{i=1}^n \|h_i^k - x^*\|^2\right],$$

where $\rho = \frac{\lambda_{\max}(\mathbf{W})}{\lambda^+_{\min}(\mathbf{W})}$, $\rho_\infty = \frac{\max_{(i,j)\in\mathcal{E}} w_{ij}}{\lambda^+_{\min}(\mathbf{W})}$, $\lambda_{\max}(\mathbf{W})$ and $\lambda^+_{\min}(\mathbf{W})$ denote the largest and smallest positive eigenvalues of $\mathbf{W}$ respectively. One can show that the factor $\rho_\infty \rho^{-1} \leq 1$ can be as small as $\Theta\left(\frac{1}{n}\right)$.

Complexity Results

Option A/B

For any given $\epsilon > 0$ **Algorithm 1 (Option A/B)** reaches accuracy $\|x - x^*\|^2 \leq \epsilon$ after the following number of iterations:

$$\mathcal{O}\left((\omega + \kappa(\rho + \omega\rho_\infty)) \log \frac{1}{\epsilon}\right).$$

Option C

For any given $\epsilon > 0$ **Algorithm 1 (Option C)** reaches accuracy $\|x - x^*\|^2 \leq \epsilon$ after the following number of iterations:

$$\tilde{\mathcal{O}}\left(\omega + (\rho + \omega\rho_\infty)\left(\kappa + \frac{\sigma\sqrt{1+\omega}}{\sqrt{\epsilon}\mu} + \frac{\sigma^2(\rho + \omega\rho_\infty)}{\epsilon\mu^2}\right)\right).$$

Option D

For any given $\epsilon > 0$ **Algorithm 1 (Option D)** reaches accuracy $\|x - x^*\|^2 \leq \epsilon$ after the following number of iterations:

$$\mathcal{O}\left((m + \omega + \kappa(\rho + \omega\rho_\infty)) \log \frac{1}{\epsilon}\right).$$

Experiments

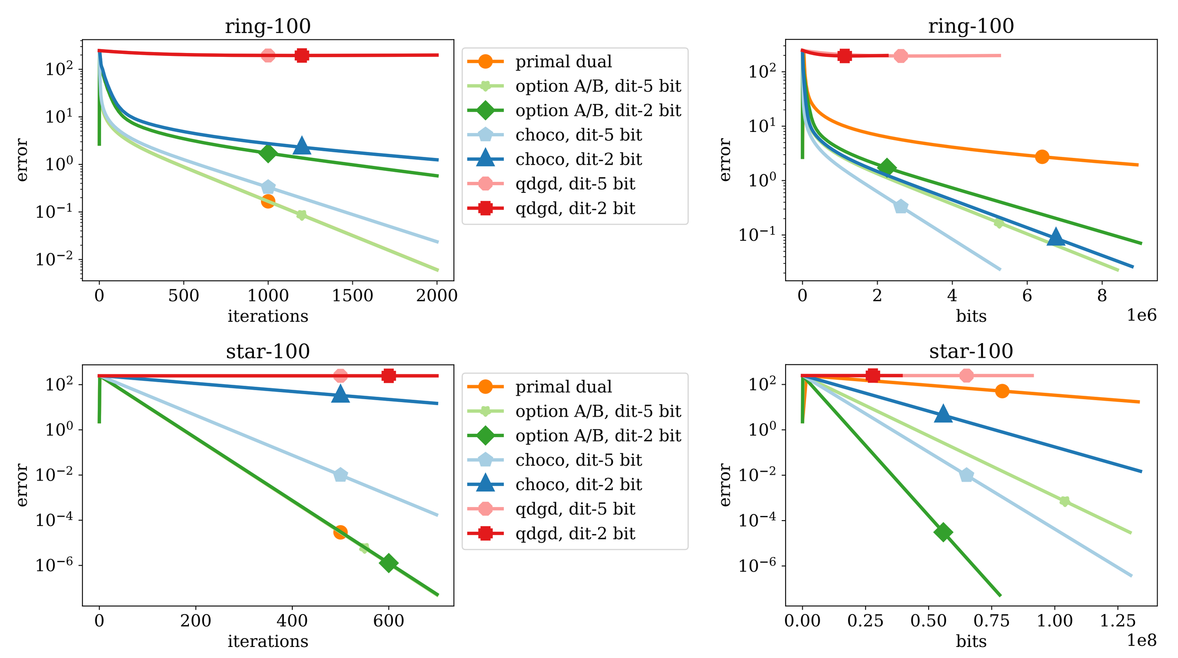

Figure 2: Comparison with the baselines: QDGD [4], Primal Dual GD [5], Choco-SGD [6]. Average consensus problem on the star and ring topologies with $n = 100$ nodes, $d = 250$, random sparsification and random dithering compression.