

A Communication Compression Decentralized Algorithm for Convex Composite Optimization

Yao Li

<https://yaoleoli.github.io/>

Joint work with Xiaorui Liu, Jiliang Tang, Ming Yan and Kun Yuan

Department of Mathematics
Department of Computational Mathematics, Science and Engineering
Michigan State University

MOPTA 2021, August 3

Outline

- 1 Problem setting
- 2 Communication compression
- 3 Proposed algorithms
 - Smooth case: LEAD
 - General case: Prox-LEAD
- 4 Convergence results
- 5 Numerical results
- 6 Conclusion

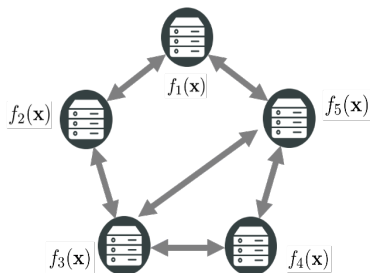
Introduction

Convex composite problem

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (f_i(\mathbf{x}) + r(\mathbf{x})) \quad (1)$$

$f_i(\cdot)$ is proper, convex and differentiable.

$r(\cdot)$ is the shared convex nonsmooth regularizer.



All agents form an undirected and connected graph.

$f_i(\cdot)$ is privately known by agent i .

Only accessible neighbors can communicate along edges.

Figure: Communication network

Introduction

Each agent has a copy \mathbf{x}_i and work collaboratively with neighbors to find the optimal solution.

Matrix notations

$$\mathbf{X}^k = \begin{bmatrix} \text{---} & (\mathbf{x}_1^k)^\top & \text{---} \\ & \vdots & \\ \text{---} & (\mathbf{x}_n^k)^\top & \text{---} \end{bmatrix} \in \mathbb{R}^{n \times p},$$
$$\nabla \mathbf{F}(\mathbf{X}^k) = \begin{bmatrix} \text{---} & (\nabla f_1(\mathbf{x}_1^k))^\top & \text{---} \\ & \vdots & \\ \text{---} & (\nabla f_n(\mathbf{x}_n^k))^\top & \text{---} \end{bmatrix} \in \mathbb{R}^{n \times p},$$

Mixing matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$ is symmetric and encodes the communication weights.

$$\mathbf{W}\mathbf{X} = \mathbf{X} \quad \text{iff} \quad \mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n,$$

$$-1 < \lambda_n(\mathbf{W}) \leq \lambda_{n-1}(\mathbf{W}) \leq \cdots \leq \lambda_2(\mathbf{W}) < \lambda_1(\mathbf{W}) = 1.$$

We have **decentralized consensus problem** (DCP)

$$\mathbf{X}^* = \arg \min_{\mathbf{X} \in \mathbb{R}^{n \times p}} \underbrace{\sum_{i=1}^n f_i(\mathbf{x}_i)}_{=: \mathbf{F}(\mathbf{X})} + \underbrace{\sum_{i=1}^n r(\mathbf{x}_i)}_{=: \mathbf{R}(\mathbf{X})}, \quad \text{s.t. } (\mathbf{I} - \mathbf{W})\mathbf{X} = \mathbf{0}, \quad (2)$$

Consensus means, in optimality

$$(\mathbf{I} - \mathbf{W})\mathbf{X}^* = \mathbf{0} \quad \Rightarrow \quad \mathbf{X}^* = \mathbf{1}(\mathbf{x}^*)^\top$$

The communication process can be formulated by $\mathbf{X}^+ = \mathbf{W}\mathbf{X}$.

In agent i 's perspective,

$$\mathbf{x}_i^+ = w_{ii}\mathbf{x}_i + \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_j.$$

Communication compression

Compress the transmitted vector in communication, e.g.,

$$[1.2, -0.1] \Rightarrow ([1, 0], \|[1.2, -0.1]\|_1)$$

Q: Why do we compress the communication?

A: The limited communication bandwidth impacts the time spent on training large models significantly.

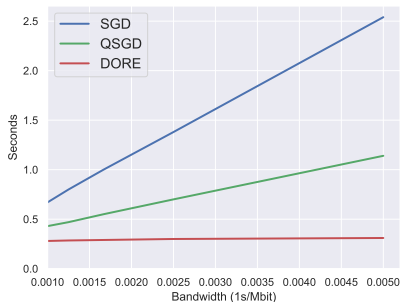


Figure: Per iteration time cost on Resnet18 for SGD, QSGD, and DORE. The figure is from [LLTY20].

Compression operator

Unbiased stochastic compression operator $\mathcal{Q} : \mathbb{R}^p \rightarrow \mathbb{R}^p$ with bounded variance-to-signal ratio, i.e.

$$\begin{aligned}\mathbb{E}\mathcal{Q}(\mathbf{x}) &= \mathbf{x}, \\ \mathbb{E}\|\mathbf{x} - \mathcal{Q}(\mathbf{x})\|^2 &\leq C\|\mathbf{x}\|^2.\end{aligned}$$

$C \geq 0$ measures the level of compression.

Examples:

p -norm b -bit quantization,

$$Q_p(\mathbf{x}) := \left(\|\mathbf{x}\|_p \text{sign}(\mathbf{x}) 2^{-(b-1)} \right) \cdot \left\lfloor \frac{2^{b-1}|\mathbf{x}|}{\|\mathbf{x}\|_p} + \mathbf{u} \right\rfloor, \quad \mathbf{u} \sim \text{Unif}[0, 1]^p.$$

random- k sparsification,

pick k elements randomly and scale for unbiasedness.

Smooth case: LEAD

Many compression algorithms have been proposed such as QDGD, QuanTimed-DSGD [RMHP19, RTM⁺19], Choco-sgd [KSJ19] and LessBit [KKJ⁺21].

We propose LEAD [LLW⁺21] with faster convergence rate and better convergence complexity.

Consider the equivalent min-max problem

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times p}} \max_{\mathbf{S} \in \mathbb{R}^{n \times p}} \mathbf{F}(\mathbf{X}) + \langle \mathbf{B}^{\frac{1}{2}} \mathbf{X}, \mathbf{S} \rangle, \quad (3)$$

where $\mathbf{B} = \frac{\mathbf{I} - \mathbf{W}}{2}$.

We apply primal-dual hybrid gradient method (PDHG) in [ZC08].

PDHG :

$$\mathbf{X}^{k+1} = \arg \min_{\mathbf{X} \in \mathbb{R}^{n \times p}} \mathbf{F}(\mathbf{X}) + \langle \mathbf{B}^{\frac{1}{2}} \mathbf{X}, \mathbf{S}^k \rangle,$$

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \lambda \mathbf{B}^{\frac{1}{2}} \mathbf{X}^{k+1}.$$

We solve \mathbf{X} -subproblem inexactly by two-step gradient descent with stepsize η .

inexact PDHG :

$$\mathbf{X}^{k+1} = \mathbf{X}^k - \eta \mathbf{F}(\mathbf{X}^k) - \eta \mathbf{B}^{\frac{1}{2}} \mathbf{S}^k,$$

$$\bar{\mathbf{X}}^{k+1} = \mathbf{X}^{k+1} - \eta \nabla \mathbf{F}(\mathbf{X}^{k+1}) - \eta \mathbf{B}^{\frac{1}{2}} \mathbf{S}^k,$$

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \lambda \mathbf{B}^{\frac{1}{2}} \bar{\mathbf{X}}^{k+1}.$$

Switch the order and let $\mathbf{D} = \mathbf{B}^{\frac{1}{2}}\mathbf{S}$.

$$\left[\begin{array}{l} \mathbf{inexact\ PDHG} : \\ \bar{\mathbf{X}}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k) - \eta \mathbf{D}^k, \\ \mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\lambda}{2}(\mathbf{I} - \mathbf{W})\bar{\mathbf{X}}^{k+1}, \\ \mathbf{X}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k) - \eta \mathbf{D}^{k+1}. \end{array} \right. \quad (4)$$

There is only one time communication in \mathbf{D} step.

We propose a new compression procedure for communication over decentralized networks.

Communication procedure

Suppose we transmit \mathbf{Y} via $(\mathbf{I} - \mathbf{W})\mathbf{Y}$, the compression estimator is generated from the following procedure.

Compressed communication procedure (**COMM**):

$$\mathbf{Q}^k = \mathcal{Q}(\mathbf{Y}^k - \mathbf{H}^k) \quad \triangleright \text{Compression}$$

$$\hat{\mathbf{Y}}^k = \mathbf{H}^k + \mathbf{Q}^k$$

$$\hat{\mathbf{Y}}_w^k = \mathbf{H}_w^k + \mathbf{W}\mathbf{Q}^k \quad \triangleright \text{Communication}$$

$$\mathbf{H}^{k+1} = (1 - \alpha)\mathbf{H}^k + \alpha\hat{\mathbf{Y}}^k$$

$$\mathbf{H}_w^{k+1} = (1 - \alpha)\mathbf{H}_w^k + \alpha\hat{\mathbf{Y}}_w^k$$

The estimator $\hat{\mathbf{Y}} - \hat{\mathbf{Y}}_w = (\mathbf{I} - \mathbf{W})\hat{\mathbf{Y}}$ is unbiased and used in algorithm instead.

Algorithm 1 LEAD

Input: Stepsize η , parameter (α, γ) , \mathbf{X}^0 , \mathbf{H}^1 , $\mathbf{D}^1 = (\mathbf{I} - \mathbf{W})\mathbf{Z}$ for any \mathbf{Z}

Output: \mathbf{X}^K or $1/n \sum_{i=1}^n \mathbf{X}_i^K$

1: $\mathbf{H}_w^1 = \mathbf{W}\mathbf{H}^1$

2: $\mathbf{X}^1 = \mathbf{X}^0 - \eta \nabla \mathbf{F}(\mathbf{X}^0; \xi^0)$

3: **for** $k = 1, 2, \dots, K - 1$ **do**

4: $\mathbf{Y}^k = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k; \xi^k) - \eta \mathbf{D}^k$

5: $\hat{\mathbf{Y}}^k, \hat{\mathbf{Y}}_w^k, \mathbf{H}^{k+1}, \mathbf{H}_w^{k+1} = \text{COMM}(\mathbf{Y}^k, \mathbf{H}^k, \mathbf{H}_w^k)$

6: $\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\gamma}{2\eta}(\hat{\mathbf{Y}}^k - \hat{\mathbf{Y}}_w^k)$

7: $\mathbf{X}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k; \xi^k) - \eta \mathbf{D}^{k+1}$

8: **end for**

Smooth case: LEAD

Each f_i is L -smooth and μ -strongly convex. $\kappa_f = \frac{L}{\mu}$, $\kappa_g = \frac{\lambda_{\max}(\mathbf{I}-\mathbf{W})}{\lambda_{\min}(\mathbf{I}-\mathbf{W})}$.

Theorem (Complexity with full gradient)

Taking the fixed stepsize, LEAD converges to the ϵ -accurate solution with the iteration complexity

$$\mathcal{O}\left(\left((1+C)(\kappa_f + \kappa_g) + C\kappa_f\kappa_g\right) \log \frac{1}{\epsilon}\right).$$

When $C = 0$ (i.e., no compression) or $C \leq \frac{\kappa_f + \kappa_g}{\kappa_f\kappa_g + \kappa_f + \kappa_g}$, the iteration complexity $\mathcal{O}\left((\kappa_f + \kappa_g) \log \frac{1}{\epsilon}\right)$ recovers the convergence rate of NIDS [LSY19].

Furthermore, when the network is fully connected, i.e., $\kappa_g = 1$, the complexity $\mathcal{O}\left(\kappa_f \log \frac{1}{\epsilon}\right)$ recovers the complexity of gradient descent [Nes13].

General case: Prox-LEAD

Return to the general problem with regularizer, we consider the min-max problem,

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times p}} \max_{\mathbf{S} \in \mathbb{R}^{n \times p}} \mathbf{F}(\mathbf{X}) + \langle \mathbf{B}^{\frac{1}{2}} \mathbf{X}, \mathbf{S} \rangle + \mathbf{R}(\mathbf{X}). \quad (5)$$

We adapt inexact PDHG with an additional proximal gradient step to have

$$\left[\begin{array}{l} \bar{\mathbf{X}}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k) - \eta \mathbf{D}^k, \\ \mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\lambda}{2} (\mathbf{I} - \mathbf{W}) \bar{\mathbf{X}}^{k+1}, \\ \mathbf{V}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k) - \eta \mathbf{D}^{k+1} = \left(\mathbf{I} - \frac{\eta \lambda}{2} (\mathbf{I} - \mathbf{W}) \right) \bar{\mathbf{X}}^{k+1}, \\ \mathbf{X}^{k+1} = \text{prox}_{\eta \mathbf{R}}(\mathbf{V}^{k+1}). \end{array} \right. \quad (6)$$

$$\text{prox}_{\eta \mathbf{R}}(\mathbf{X}) = \arg \min_{\mathbf{Y} \in \mathbb{R}^{n \times p}} \mathbf{R}(\mathbf{Y}) + \frac{1}{2\eta} \|\mathbf{Y} - \mathbf{X}\|^2.$$

Convergence of Prox-LEAD

Theorem (Complexity with full gradient)

Under the same assumptions as LEAD, Prox-LEAD converges to the ϵ -accurate solution with the iteration complexity

$$\mathcal{O}\left(\left((1 + C)(\kappa_f + \kappa_g) + \sqrt{C}(1 + C)\kappa_f\kappa_g\right) \log \frac{1}{\epsilon}\right).$$

For stochastic gradient, we consider two different settings.

The general stochastic setting:

$$f_i(\mathbf{x}_i) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i} f_i(\mathbf{x}_i, \xi_i).$$

The finite-sum setting:

$$f_i(\mathbf{x}_i) = \frac{1}{m} \sum_{j=1}^m f_{ij}(\mathbf{x}_i).$$

Prox-LEAD is compatible with variance reduction schemes in finite sum setting.

Convergence of Prox-LEAD

In the general stochastic setting, we assume each f_i is L -smooth in expectation and μ -strongly convex.

The local stochastic gradient satisfies

$$\mathbb{E}\nabla f_i(\mathbf{x}_i, \xi_i) = \nabla f_i(\mathbf{x}_i).$$

▷ unbiasedness

$$\mathbb{E}\|\nabla f_i(\mathbf{x}^*, \xi_i) - \nabla f_i(\mathbf{x}^*)\|^2 \leq \sigma_i^2.$$

▷ bounded local variance

Theorem (Convergence rate)

Taking the fixed stepsize, the sequence $\{\mathbf{X}^k\}$ generated by Prox-LEAD satisfies

$$\mathbb{E}\|\mathbf{X}^k - \mathbf{X}^*\|^2 \leq (1 - \rho)^k M + \mathcal{O}(\sigma^2)$$

where $M > 0$, $\sigma^2 = \frac{1}{n} \sum_{i=1}^n \sigma_i^2$ and $\rho =$

$$\left(\max \left\{ 48\sqrt{C}(1+C)\kappa_f\kappa_g, 12(1+C)\kappa_f, \frac{282\kappa_f}{23}, 48(1+C)\kappa_g \right\} \right)^{-1}.$$

Convergence of Prox-LEAD

Theorem (Complexity with diminishing stepsize)

Taking the diminishing stepsize, Prox-LEAD converges to the ϵ -accurate solution with the iteration complexity

$$\mathcal{O}\left(\left((1 + C)^2 \kappa_f \kappa_g + \frac{\sigma^2}{L^2} (1 + C)^4 \kappa_f^2 \kappa_g^2\right) \frac{1}{\epsilon}\right).$$

Prox-LEAD can be accelerated to have global linear convergence with the fixed stepsize by variance reduction schemes if the problem is finite-sum.

e.g., Loopless-SVRG [KHR20] and SAGA [DBLJ14]

Convergence of Prox-LEAD

In finite sum setting, we assume each local objective function on minibatch f_{ij} is L -smooth and μ -strongly convex.

Theorem (Convergence complexity of Prox-LEAD SAGA)

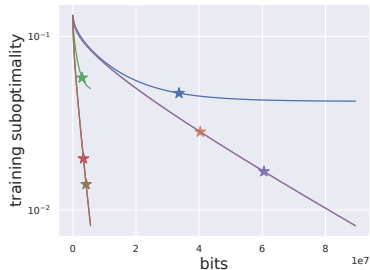
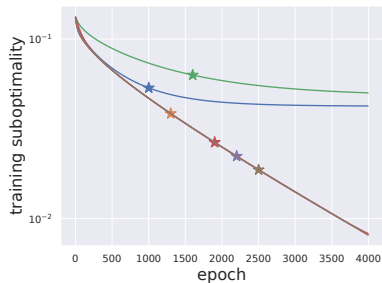
Taking the fixed stepsizes, Prox-LEAD SAGA converges to the ϵ -accurate solution with the iteration complexity

$$\mathcal{O}\left((1 + C)(\kappa_f + \kappa_g) + \sqrt{C}(1 + C)\kappa_f\kappa_g + m\right) \log \frac{1}{\epsilon}.$$

When $C = 0$, the complexity is reduced to $\mathcal{O}\left((\kappa_f + \kappa_g + m) \log \frac{1}{\epsilon}\right)$.

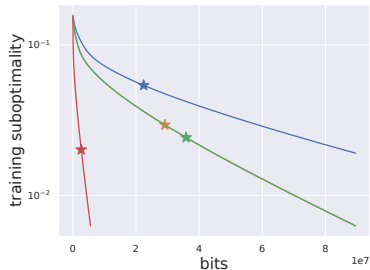
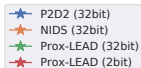
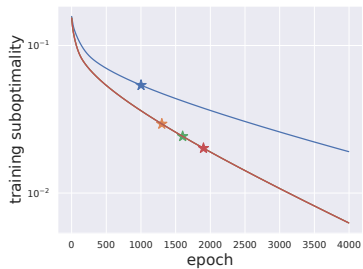
Furthermore, when $\kappa_g = 1$, the complexity $\mathcal{O}\left((\kappa_f + m) \log \frac{1}{\epsilon}\right)$ recovers the complexity of SAGA.

Experiment: logistic regression



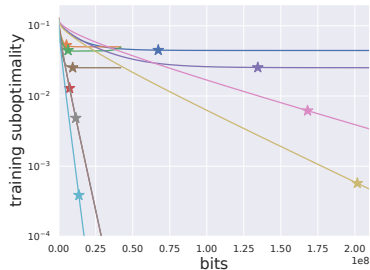
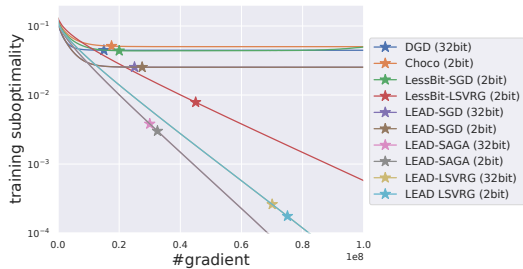
ℓ_2 regularizer with full gradient

Experiment: logistic regression



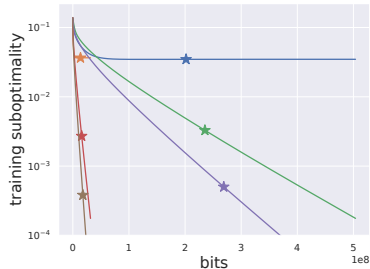
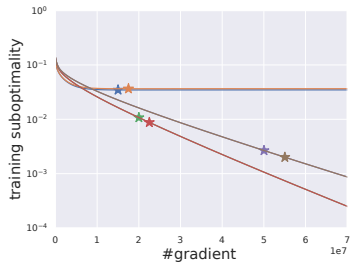
$\ell_2 + \ell_1$ regularizer with full gradient

Experiment: logistic regression



ℓ_2 regularizer with stochastic gradient

Experiment: logistic regression




$\ell_2 + \ell_1$ regularizer with stochastic gradient


Take-home message


1. Prox-LEAD is the first primal-dual stochastic algorithm with compressed communication for decentralized composite optimization and achieves linear convergence with full gradient.
2. Prox-LEAD can be combine with Loopless-SVRG and SAGA to achieve exact linear convergence with stochastic gradient.
3. Prox-LEAD doesn't require bounded assumption on data heterogeneity. Prox-LEAD is robust to parameter tuning.
4. The communication compression procedure, **COMM**, and the unbiased compression operator can be applied to other decentralized algorithms to have efficient communication.

-  Aaron Defazio, Francis Bach, and Simon Lacoste-Julien, *Saga: A fast incremental gradient method with support for non-strongly convex composite objectives*, arXiv preprint arXiv:1407.0202 (2014).
-  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*, Algorithmic Learning Theory, PMLR, 2020, pp. 451–467.
-  Dmitry Kovalev, Anastasia Koloskova, Martin Jaggi, Peter Richtarik, and Sebastian Stich, *A linearly convergent algorithm for decentralized optimization: Sending less bits for free!*, Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Arindam Banerjee and Kenji Fukumizu, eds.), Proceedings of Machine Learning Research, vol. 130, PMLR, 13–15 Apr 2021, pp. 4087–4095.
-  Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi, *Decentralized stochastic optimization and gossip algorithms with*


compressed communication, Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 3479–3487.

 Xiaorui Liu, Yao Li, Jiliang Tang, and Ming Yan, *A double residual compression algorithm for efficient distributed learning*, International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 133–143.

 Xiaorui Liu, Yao Li, Rongrong Wang, Jiliang Tang, and Ming Yan, *Linear convergent decentralized optimization with compression*, International Conference on Learning Representations, 2021.

 Zhi Li, Wei Shi, and Ming Yan, *A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates*, IEEE Transactions on Signal Processing **67** (2019), no. 17, 4494–4506.

 Yurii Nesterov, *Introductory lectures on convex optimization: A basic course*, vol. 87, Springer Science & Business Media, 2013.

 Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani, *An exact quantized decentralized gradient descent*

algorithm, IEEE Transactions on Signal Processing **67** (2019), no. 19, 4934–4947.



Amirhossein Reisizadeh, Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani, *Robust and communication-efficient collaborative learning*, Advances in Neural Information Processing Systems, 2019, pp. 8388–8399.



Mingqiang Zhu and Tony Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA CAM Report **34** (2008), 8–34.

Thank You!