# Linear Convergent Decentralized Optimization with Compression

**Xiaorui Liu**
http://cse.msu.edu/~xiaorui/

Joint work with Yao Li, Rongrong Wang,
Jiliang Tang, and Ming Yan

Data Science and Engineering Lab
Department of Computer Science and Engineering
Michigan State University

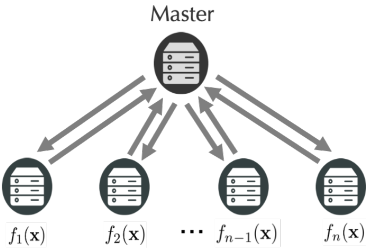
ICLR 2021, May 6th

# Introduction

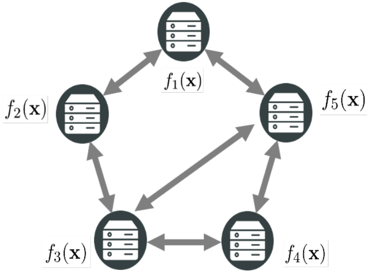
- Problem

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

- $f_i(\cdot)$ is the local objective in agent $i$.




Centralization




Decentralization

# Introduction

- Matrix notations

$$\mathbf{X}^k = \begin{bmatrix} — & (\mathbf{x}_1^k)^\top & — \\ & \vdots & \\ — & (\mathbf{x}_n^k)^\top & — \end{bmatrix} \in \mathbb{R}^{n \times d},$$

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

- Symmetric $\mathbf{W} \in \mathbb{R}^{n \times n}$ encodes the communication network.

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

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

# Introduction

- Communication Compression for decentralized optimization
  - DCD-SGD, ECE-SGD [TGZ+18]
  - QDGD, QuanTimed-DSGD[RMHP19, RTM+19]
  - DeepSqueeze [TLQ+19]
  - CHOCO-SGD [KSJ19]
  - ...
- Reduce to DGD-type algorithms, which suffer from convergence bias

$$\mathbf{X}^* \neq \mathbf{W}\mathbf{X}^* - \eta \nabla \mathbf{F}(\mathbf{X}^*).$$

  Their convergences degrade on heterogeneous data.
- LEAD is the first primal-dual decentralized optimization algorithm with compression and attains linear convergence.

# Algorithm: LEAD

- NIDS [LSY19] / D$^2$ [TLY$^+$18] (stochastic version of NIDS)

$$\mathbf{X}^{k+1} = \frac{\mathbf{I}+\mathbf{W}}{2}(2\mathbf{X}^k - \mathbf{X}^{k-1} - \eta\nabla\mathbf{F}(\mathbf{X}^k;\xi^k) + \eta\nabla\mathbf{F}(\mathbf{X}^{k-1};\xi^{k-1})),$$

- A two step reformulation [LY19]:

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\mathbf{I}-\mathbf{W}}{2\eta}(\mathbf{X}^k - \eta\nabla\mathbf{F}(\mathbf{X}^k) - \eta\mathbf{D}^k),$$
$$\mathbf{X}^{k+1} = \mathbf{X}^k - \eta\nabla\mathbf{F}(\mathbf{X}^k) - \eta\mathbf{D}^{k+1},$$

- Concise and conceptual form of LEAD:

$$\mathbf{Y}^k = \mathbf{X}^k - \eta\nabla\mathbf{F}(\mathbf{X}^k;\xi^k) - \eta\mathbf{D}^k$$
$$\hat{\mathbf{Y}}^k = \mathit{CompressionProcedure}(\mathbf{Y}^k)$$
$$\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\gamma}{2\eta}(\mathbf{I}-\mathbf{W})\hat{\mathbf{Y}}^k$$
$$\mathbf{X}^{k+1} = \mathbf{X}^k - \eta\nabla\mathbf{F}(\mathbf{X}^k;\xi^k) - \eta\mathbf{D}^{k+1}$$

# Algorithm: LEAD

- LEAD

$$\mathbf{Y}^k = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k; \xi^k) - \eta \mathbf{D}^k$$
$$\hat{\mathbf{Y}}^k = \mathit{CompressionProcedure}(\mathbf{Y}^k)$$
$$\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\gamma}{2\eta}(\mathbf{I} - \mathbf{W})\hat{\mathbf{Y}}^k = \frac{\gamma}{2\eta}(\hat{\mathbf{Y}}^k - \hat{\mathbf{Y}}^k_w)$$
$$\mathbf{X}^{k+1} = \mathbf{X}^k - \eta \nabla \mathbf{F}(\mathbf{X}^k; \xi^k) - \eta \mathbf{D}^{k+1}$$

- Compression Procedure

$$\mathbf{Q}^k = \text{Compress}(\mathbf{Y}^k - \mathbf{H}^k) \quad \triangleright \textit{Compression}$$
$$\hat{\mathbf{Y}}^k = \mathbf{H}^k + \mathbf{Q}^k$$
$$\hat{\mathbf{Y}}^k_w = \mathbf{H}^k_w + \mathbf{W}\mathbf{Q}^k \quad \triangleright \textit{Communication}$$
$$\mathbf{H}^{k+1} = (1-\alpha)\mathbf{H}^k + \alpha \hat{\mathbf{Y}}^k$$
$$\mathbf{H}^{k+1}_w = (1-\alpha)\mathbf{H}^k_w + \alpha \hat{\mathbf{Y}}^k_w$$

# Algorithm: LEAD

How LEAD works?

- Gradient Correction

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

$$\mathbf{F}(\mathbf{X}^k; \xi^k) + \mathbf{D}^{k+1} \to \mathbf{0}$$

- Difference Compression

$$\mathbf{Q}^k = \text{Compress}(\mathbf{Y}^k - \mathbf{H}^k)$$

$$\mathbf{Y}^k \to \mathbf{X}^*, \mathbf{H}^k \to \mathbf{X}^* \Rightarrow \mathbf{Y}^k - \mathbf{H}^k \to \mathbf{0} \Rightarrow \|\mathbf{Q}^k - (\mathbf{Y}^k - \mathbf{H}^k)\| \to 0$$

- Implicit Error Compensation

$$\mathbf{E}^k = \hat{\mathbf{Y}}^k - \mathbf{Y}^k$$

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\gamma}{2\eta}(\hat{\mathbf{Y}}^k - \hat{\mathbf{Y}}^k_w) = \mathbf{D}^k + \frac{\gamma}{2\eta}(\mathbf{I} - \mathbf{W})\mathbf{Y}^k + \frac{\gamma}{2\eta}(\mathbf{E}^k - \mathbf{W}\mathbf{E}^k)$$

# Assumption

- Compression: $\mathbb{E}Q(\mathbf{x}) = \mathbf{x}$, $\mathbb{E}\|\mathbf{x} - Q(\mathbf{x})\|_2^2 \leq C\|\mathbf{x}\|_2^2$ for some $C \geq 0$.
- $f_i(\cdot)$ is $\mu$-strongly convex and $L$-smooth:

$$f_i(\mathbf{x}) \geq f_i(\mathbf{y}) + \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2,$$
$$f_i(\mathbf{x}) \leq f_i(\mathbf{y}) + \langle \nabla f_i(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

- Gradient: $\mathbb{E}_\xi \nabla f_i(\mathbf{x}; \xi) = \nabla f_i(\mathbf{x})$, $\mathbb{E}_\xi\|\nabla f_i(\mathbf{x}; \xi) - \nabla f_i(\mathbf{x})\|_2^2 \leq \sigma^2$.

# Theory

$$\kappa_f = \frac{L}{\mu}, \quad \kappa_g = \frac{\lambda_{\max}(\mathbf{I} - \mathbf{W})}{\lambda_{\min}^{+}(\mathbf{I} - \mathbf{W})}$$

## Theorem (Complexity bounds when $\sigma = 0$)

- *LEAD converges to the $\epsilon$-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 is*

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

*This recovers the convergence rate of NIDS [LSY19].*

# Theory

**Theorem (Complexity bounds when $\sigma = 0$)**

- *With $C = 0$ (or $C \leq \frac{\kappa_f + \kappa_g}{\kappa_f \kappa_g + \kappa_f + \kappa_g}$) and fully connected communication graph (i.e., $\mathbf{W} = \frac{\mathbf{1}\mathbf{1}^\top}{n}$), the iteration complexity is*

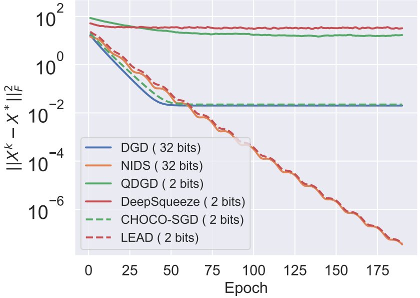
$$\mathcal{O}(\kappa_f \log \frac{1}{\epsilon}).$$

*This recovers the convergence rate of gradient descent [Nes13].*

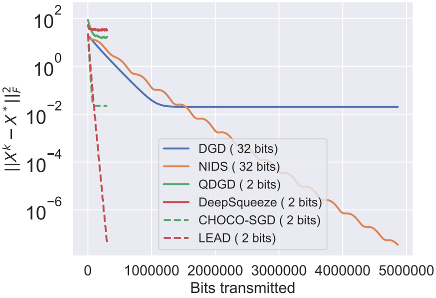
**Theorem (Error bound when $\sigma > 0$)**

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left\|\mathbf{x}_i^k - \mathbf{x}^*\right\|^2 \lesssim \mathcal{O}\left(\frac{1}{k}\right)$$

# Experiment



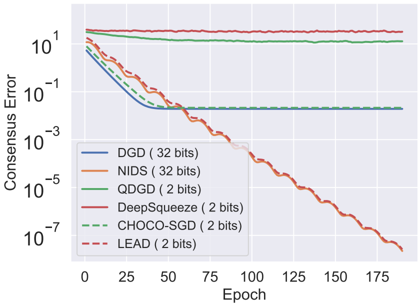

$\|\mathbf{X}^k - \mathbf{X}^*\|_F$



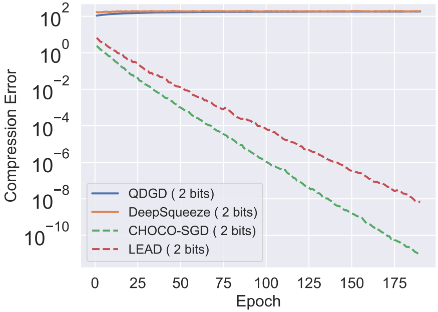

$\|\mathbf{X}^k - \mathbf{X}^*\|_F$

Linear regression $(\sigma = 0)$

# Experiment



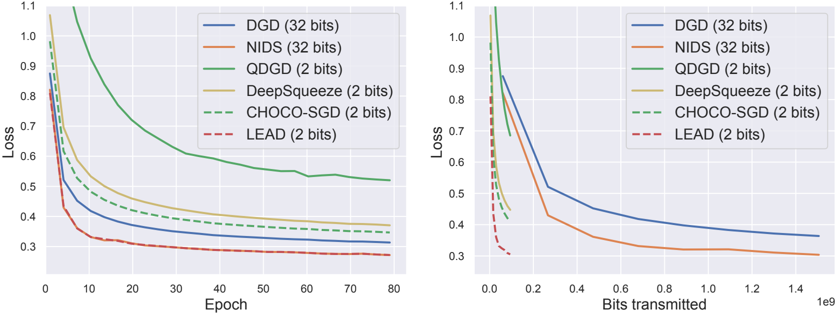

Consensus error




Compression error

Linear regression ($\sigma = 0$)

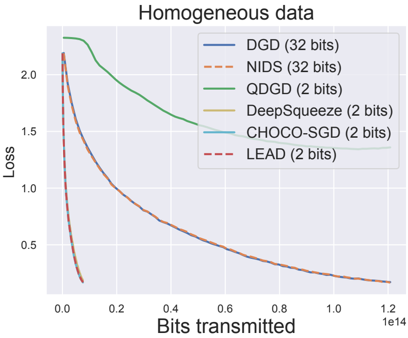
# Experiment



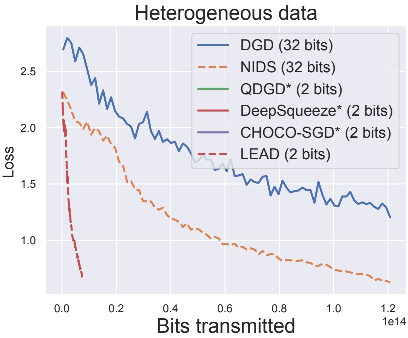

$f(\overline{\mathbf{X}}^k)$ $f(\overline{\mathbf{X}}^k)$

Logistic regression ($\sigma > 0$).

# Experiment




Loss $f(\overline{\mathbf{X}}^k)$




Loss $f(\overline{\mathbf{X}}^k)$

Stochastic optimization on deep learning ($*$ divergence).

# Conclusion

- LEAD is the first primal-dual decentralized optimization algorithm with compression and attains linear convergence for strongly convex and smooth objectives
- LEAD supports unbiased compression of arbitrary precision
- LEAD works well for nonconvex problems such as training deep neural networks
- LEAD is robust to parameter settings, and needs minor effort for parameter tuning

Welcome to check our paper and poster for more details

- 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.
- 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.
- Yao Li and Ming Yan, *On linear convergence of two decentralized algorithms*, arXiv preprint arXiv:1906.07225 (2019).
- 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.
- Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu, *Communication compression for decentralized training*, Advances in Neural Information Processing Systems, 2018, pp. 7652–7662.
- Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu, *Deepsqueeze: Decentralization meets error-compensated compression*, CoRR **abs/1907.07346** (2019).
- Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu, *$D^2$: Decentralized training over decentralized data*, Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 4848–4856.