BEER: Fast $O(1/T)$ Rate for Decentralized Nonconvex Optimization with Communication Compression

Zhize Li
Carnegie Mellon University
https://zhizeli.github.io

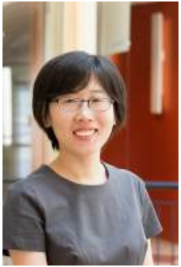
May 2, 2022

Joint work with

Haoyu Zhao

Boyue Li

Peter Richtárik



Yuejie Chi

Overview

Optimization Problem

We consider the decentralized optimization problem:

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

x: model parameters,
n: number of clients,
$f_i(x)$: loss function on client i, $f_i(x) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i} f(x;\xi_i)$, where $\mathcal{D}_i$ is the local dataset on client i.

Note that each client can only communicate with its neighbors via a predefined network topology (captured by a mixing matrix W).

Challenges

There are many challenges in decentralized optimization:

- High communication cost
- Heterogeneous/Non-IID data, the data distribution $\mathcal{D}_i$ may vary from different clients
- Data privacy
- ...

We will focus on the **communication cost** and **heterogeneous data**.

Related Work

To reduce communication cost, people usually use **compressed communication** (e.g., Alistarh et al. (2017); Stich et al. (2018); Koloskova et al. (2019); Richtárik et al. (2021)).

Definition (compression operator)

A randomized map $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an α-compression operator if for all $x \in \mathbb{R}^d$, it satisfies

$$\mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq (1 - \alpha)\|x\|^2. \tag{2}$$

In particular, no compression ($\mathcal{C}(x) \equiv x$) implies $\alpha = 1$.

Related Work

To reduce communication cost, people usually use **compressed communication** (e.g., Alistarh et al. (2017); Stich et al. (2018); Koloskova et al. (2019); Richtárik et al. (2021)).

Definition (compression operator)

A randomized map $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an α-compression operator if for all $x \in \mathbb{R}^d$, it satisfies

$$\mathbb{E}[\|\mathcal{C}(x) - x\|^2] \leq (1 - \alpha)\|x\|^2. \tag{2}$$

In particular, no compression ($\mathcal{C}(x) \equiv x$) implies $\alpha = 1$.

Examples: $\mathbf{random}_k(x) = x \odot u$ (where u is a uniformly random binary vector with k nonzero entries, $\odot$ denotes element-wise product) satisfies (2) with $\boldsymbol{\alpha = k/d}$. $\mathbf{top}_k(x)$ also satisfies (2) with $\boldsymbol{\alpha = k/d}$.

Related Work

Although previous works reduce the communication cost via compression, they achieve **slow convergence rates** (need more communication rounds) and require **bounded gradient/dissimilarity assumption** (do not suit for heterogeneous data setting)

Related Work

Although previous works reduce the communication cost via compression, they achieve **slow convergence rates** (need more communication rounds) and require **bounded gradient/dissimilarity assumption** (do not suit for heterogeneous data setting)

Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$, where $f_i(x) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i} f(x;\xi_i)$, and $\mathcal{D}_i$ is the local dataset on client i.

Related Work

Although previous works reduce the communication cost via compression, they achieve **slow convergence rates** (need more communication rounds) and require **bounded gradient/dissimilarity assumption** (do not suit for heterogeneous data setting)

Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right\}$, where $f_i(x) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i} f(x;\xi_i)$, and $\mathcal{D}_i$ is the local dataset on client i.

- **Bounded gradient:** $\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\|\nabla f(x;\xi_i)\|^2 \le G^2$
- **Bounded dissimilarity:** $\mathbb{E}_i\|\nabla f_i(x) - \nabla f(x)\|^2 \le G^2$

Result Comparison

Table: Decentralized nonconvex optimization with communication compression

Algorithm	Convergence rate	Strong assumption
SQuARM-SGD (Singh et al., 2021)	$O\left(\frac{1}{\sqrt{nT}} + \frac{nG^2}{T}\right)$	Bounded Gradient
DeepSqueeze (Tang et al., 2019)	$O\left(\left(\frac{G}{T}\right)^{2/3}\right)$	Bounded Dissimilarity
CHOCO-SGD (Koloskova et al., 2019)	$O\left(\left(\frac{G}{T}\right)^{2/3}\right)$	Bounded Gradient
BEER (this paper)	$O\left(\frac{1}{T}\right)$	–

***T*:** number of communication rounds

***n*:** total number of clients

***G*:** bounded gradient/dissimilarity assumption

$$(\mathbb{E}_{\xi_i \sim \mathcal{D}_i}\|\nabla f(x;\xi_i)\|^2 \le G^2 \text{ or } \mathbb{E}_i\|\nabla f_i(x) - \nabla f(x)\|^2 \le G^2)$$

Our Approaches

CHOCO-SGD (Koloskova et al., 2019): $O\left(\left(\frac{G}{T}\right)^{2/3}\right)$ vs. BEER: $O\left(\frac{1}{T}\right)$

- Improving $O(1/T^{2/3})$ to $O(1/T)$:

CHOCO-SGD uses the original Error Feedback (**EF**) compression framework (Seide et al., 2014), while **BEER** adopts a better **EF21** compression framework (Richtárik et al., 2021).

Our Approaches

CHOCO-SGD (Koloskova et al., 2019): $O\left(\left(\frac{G}{T}\right)^{2/3}\right)$ vs. BEER: $O\left(\frac{1}{T}\right)$

- Improving $O(1/T^{2/3})$ to $O(1/T)$:

CHOCO-SGD uses the original Error Feedback (**EF**) compression framework (Seide et al., 2014), while **BEER** adopts a better **EF21** compression framework (Richtárik et al., 2021).

- Removing bounded gradient/dissimilarity G:

CHOCO-SGD uses **plain gradients**, while **BEER** adopts the **gradient tracking** idea (Zhu and Martínez (2010); Nedić et al. (2017)).

Direct Compression Framework

- Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$.
- Recall the compression operator $\mathcal{C}$, s.t. $\mathbb{E}[\|\mathcal{C}(x)-x\|^2] \le (1-\alpha)\|x\|^2$.
- We point out that **direct compression framework** $x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^t))$ **does not work**.

Direct Compression Framework

- Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$.
- Recall the compression operator $\mathcal{C}$, s.t. $\mathbb{E}[\|\mathcal{C}(x)-x\|^2] \le (1-\alpha)\|x\|^2$.
- We point out that **direct compression framework**

$$x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^t)) \quad \textbf{does not work}.$$

A counter-example: consider $n=3$ and let $f_i(x) = (a_i^\top x)^2 + \frac{1}{2}\|x\|^2$, where $a_1 = (-4,3,3)^\top$, $a_2 = (3,-4,3)^\top$ and $a_3 = (3,3,-4)^\top$.

Direct Compression Framework

- Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$.
- Recall the compression operator $\mathcal{C}$, s.t. $\mathbb{E}[\|\mathcal{C}(x)-x\|^2] \le (1-\alpha)\|x\|^2$.
- We point out that **direct compression framework**

$$x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^t)) \quad \textbf{does not work}.$$

A counter-example: consider $n=3$ and let $f_i(x) = (a_i^\top x)^2 + \frac{1}{2}\|x\|^2$, where $a_1 = (-4,3,3)^\top$, $a_2 = (3,-4,3)^\top$ and $a_3 = (3,3,-4)^\top$.

If algorithm starts with $x^0 = (b,b,b)$, then $\nabla f_1(x^0) = b(-15,13,13)^\top$, $\nabla f_2(x^0) = b(13,-15,13)^\top$, and $\nabla f_3(x^0) = b(13,13,-15)^\top$.

Direct Compression Framework

- Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$.
- Recall the compression operator $\mathcal{C}$, s.t. $\mathbb{E}[\|\mathcal{C}(x)-x\|^2] \le (1-\alpha)\|x\|^2$.
- We point out that **direct compression framework**

$$x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^t)) \quad \textbf{does not work}.$$

A counter-example: consider $n=3$ and let $f_i(x) = (a_i^\top x)^2 + \frac{1}{2}\|x\|^2$, where $a_1 = (-4,3,3)^\top$, $a_2 = (3,-4,3)^\top$ and $a_3 = (3,3,-4)^\top$.

If algorithm starts with $x^0 = (b,b,b)$, then $\nabla f_1(x^0) = b(-15,13,13)^\top$, $\nabla f_2(x^0) = b(13,-15,13)^\top$, and $\nabla f_3(x^0) = b(13,13,-15)^\top$.

If the compressor is $\mathbf{top_1}$, we have $\mathcal{C}(\nabla f_1(x^0)) = b(-15,0,0)^\top$, $\mathcal{C}(\nabla f_2(x^0)) = b(0,-15,0)^\top$, $\mathcal{C}(\nabla f_3(x^0)) = b(0,0,-15)^\top$,

Direct Compression Framework

- Recall the problem here: $\min_{x\in\mathbb{R}^d}\left\{f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)\right\}$.
- Recall the compression operator $\mathcal{C}$, s.t. $\mathbb{E}[\|\mathcal{C}(x)-x\|^2] \le (1-\alpha)\|x\|^2$.
- We point out that **direct compression framework**

$$x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}(\nabla f_i(x^t)) \quad \textbf{does not work}.$$

A counter-example: consider $n=3$ and let $f_i(x) = (a_i^\top x)^2 + \frac{1}{2}\|x\|^2$, where $a_1 = (-4,3,3)^\top$, $a_2 = (3,-4,3)^\top$ and $a_3 = (3,3,-4)^\top$.

If algorithm starts with $x^0 = (b,b,b)$, then $\nabla f_1(x^0) = b(-15,13,13)^\top$, $\nabla f_2(x^0) = b(13,-15,13)^\top$, and $\nabla f_3(x^0) = b(13,13,-15)^\top$.

If the compressor is **top$_1$**, we have $\mathcal{C}(\nabla f_1(x^0)) = b(-15,0,0)^\top$, $\mathcal{C}(\nabla f_2(x^0)) = b(0,-15,0)^\top$, $\mathcal{C}(\nabla f_3(x^0)) = b(0,0,-15)^\top$,
and the next iteration $x^1 = x^0 - \eta\frac{1}{3}\sum_{i=1}^3 \mathcal{C}(\nabla f_i(x^0)) = (1+5\eta)x^0$, and then $\boldsymbol{x^t = (1+5\eta)^t x^0}$ **diverges exponentially**.

Error Feedback (EF) Compression Framework

EF was first proposed by Seide et al. (2014) as a heuristic, no theoretical understanding until recently (Stich et al. (2018); Alistarh et al. (2018)).

Each client $i \in [n]$ sets the zero initial error $e_i^0 = 0$
Each client $i \in [n]$ compress its initial gradient $\boldsymbol{g_i^0} = \mathcal{C}(\gamma \nabla f_i(x^0))$
for $t = 0, 1, 2, \ldots$ **do**
    Server updates $x^{t+1} = x^t - \frac{1}{n}\sum_{i=1}^n \boldsymbol{g_i^t}$
    for all clients $i = 1, 2, \ldots, n$ **do in parallel**
        Compute error: $e_i^{t+1} = e_i^t + \gamma \nabla f_i(x^t) - g_i^t$
        Compress error-compensated gradient g_i^{t+1} and send to server:

$$\boldsymbol{g_i^{t+1}} = \mathcal{C}(e_i^{t+1} + \gamma \nabla f_i(x^{t+1}))$$

end for

Error Feedback (EF) vs. EF21

To compare them clearly, consider the case $n = 1$ (single node):

EF (Seide et al., 2014)

1: Model update: $x^{t+1} = x^t - g^t$
2: Error: $e^{t+1} = e^t + \gamma \nabla f(x^t) - g^t$
3: Compress error-compensated gradient: $g^{t+1} = \mathcal{C}(e^{t+1} + \gamma \nabla f(x^{t+1}))$

EF21 (Richtárik et al., 2021)

1: Model update: $x^{t+1} = x^t - \gamma g^t$
2: Update with a shifted compression: $g^{t+1} = g^t + \mathcal{C}(\nabla f(x^{t+1}) - g^t)$

Error Feedback (EF) vs. EF21

To compare them clearly, consider the case $n = 1$ (single node):

EF (Seide et al., 2014)

1: Model update: $x^{t+1} = x^t - g^t$
2: Error: $e^{t+1} = e^t + \gamma \nabla f(x^t) - g^t$
3: Compress error-compensated gradient: $g^{t+1} = \mathcal{C}(e^{t+1} + \gamma \nabla f(x^{t+1}))$

EF21 (Richtárik et al., 2021)

1: Model update: $x^{t+1} = x^t - \gamma g^t$
2: Update with a shifted compression: $g^{t+1} = g^t + \mathcal{C}(\nabla f(x^{t+1}) - g^t)$

If compressor $\mathcal{C}$ is additive and positively homogeneous, **EF** $=$ **EF21**.

$$
\begin{aligned}
g^{t+1} &= \mathcal{C}(e^{t+1} + \gamma \nabla f(x^{t+1})) = \mathcal{C}(e^t + \gamma \nabla f(x^t) - g^t + \gamma \nabla f(x^{t+1})) \\
&= \mathcal{C}(e^t + \gamma \nabla f(x^t)) + \mathcal{C}(\gamma \nabla f(x^{t+1}) - g^t) = g^t + \mathcal{C}(\gamma \nabla f(x^{t+1}) - g^t).
\end{aligned}
$$

Let g^t denote $\gamma \hat{g}^t$, then $g^{t+1} = \gamma(\hat{g}^t + \mathcal{C}(\nabla f(x^t) - \hat{g}^t)) = \gamma \hat{g}^{t+1}$.

Recall Our Approaches

CHOCO-SGD (Koloskova et al., 2019): $O\left(\left(\frac{G}{T}\right)^{2/3}\right)$ vs. BEER: $O\left(\frac{1}{T}\right)$

- Improving $O(1/T^{2/3})$ to $O(1/T)$:

CHOCO-SGD uses the original Error Feedback (**EF**) compression framework (Seide et al., 2014), while **BEER** adopts a better **EF21** compression framework (Richtárik et al., 2021).

- Removing bounded gradient/dissimilarity G:

CHOCO-SGD uses **plain gradients**, while **BEER** adopts the **gradient tracking** idea (Zhu and Martínez (2010); Nedić et al. (2017)).

CHOCO-SGD (Koloskova et al., 2019)

Algorithm 4 CHOCO-SGD (Koloskova et al., 2019) as Error Feedback

input: Initial values $\mathbf{x}_i^{(0)} \in \mathbb{R}^d$ on each node $i \in [n]$, consensus stepsize γ, SGD stepsize η, comm. graph $G = ([n], E)$ and mixing matrix W, initialize $\hat{\mathbf{x}}_i^{(0)} = \mathbf{x}_i^{(-1)} := \mathbf{0}$, $\forall i \in [n]$

for t **in** $0 \dots T-1$ **do** {*in parallel for all workers* $i \in [n]$}

Error Feedback (EF)

$\mathbf{x}_i^{(t)} := \mathbf{x}_i^{(t-\frac{1}{2})} + \gamma \sum_{j:\{i,j\} \in E} w_{ij} \left(\hat{\mathbf{x}}_j^{(t)} - \hat{\mathbf{x}}_i^{(t)}\right)$ ◁ modified gossip averaging

$\mathbf{v}_i^{(t)} = \mathbf{x}_i^{(t)} - \mathbf{x}_i^{(t-1)} + \mathbf{m}_i^{(t)}$

$\mathbf{q}_i^{(t)} := Q(\mathbf{v}_i^{(t)})$ ◁ compression

$\mathbf{m}_i^{(t+1)} = \mathbf{v}_i^{(t)} - \mathbf{q}_i^{(t)}$ ◁ memory update

for neighbors j: $\{i,j\} \in E$ (including $\{i\} \in E$) **do**

Send $\mathbf{q}_i^{(t)}$ and receive $\mathbf{q}_j^{(t)}$ ◁ communication

$\hat{\mathbf{x}}_j^{(t+1)} := \mathbf{q}_j^{(t)} + \hat{\mathbf{x}}_j^{(t)}$ ◁ local update

end for

Sample $\xi_i^{(t)}$, compute gradient $\mathbf{g}_i^{(t)} := \nabla F_i(\mathbf{x}_i^{(t)}, \xi_i^{(t)})$ plain gradients

$\mathbf{x}_i^{(t+\frac{1}{2})} := \mathbf{x}_i^{(t)} - \eta \mathbf{g}_i^{(t)}$ ◁ stochastic gradient update

end for

Our BEER Algorithm

Algorithm 1 BEER: BEtter comprEssion for decentRalized optimization

1: **Input:** Initial point $\boldsymbol{X}^0 = \boldsymbol{x}_0 \mathbf{1}^\top$, $\boldsymbol{G}^0 = \mathbf{0}$, $\boldsymbol{H}^0 = \mathbf{0}$, $\boldsymbol{V}^0 = \nabla F(\boldsymbol{X}_0)$, step size η, mixing step size γ, minibatch size b

2: **for** $t = 0, 1, \ldots$ **do**

EF21

3: $\quad \boldsymbol{X}^{t+1} = \boldsymbol{X}^t + \gamma \boldsymbol{H}^t(\boldsymbol{W} - \boldsymbol{I}) - \eta \boldsymbol{V}^t$

4: $\quad \boldsymbol{H}^{t+1} = \boldsymbol{H}^t + \mathcal{C}(\boldsymbol{X}^{t+1} - \boldsymbol{H}^t)$

gradient tracking

5: $\quad \boldsymbol{V}^{t+1} = \boldsymbol{V}^t + \gamma \boldsymbol{G}^t(\boldsymbol{W} - \boldsymbol{I}) + \tilde{\nabla}_b F(\boldsymbol{X}^{t+1}) - \tilde{\nabla}_b F(\boldsymbol{X}^t)$

6: $\quad \boldsymbol{G}^{t+1} = \boldsymbol{G}^t + \mathcal{C}(\boldsymbol{V}^{t+1} - \boldsymbol{G}^t)$

EF21

7: **end for**

Plain Gradients vs. Gradient Tracking

Let $\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$ denote the collection of parameters from all clients, and $\nabla F(\boldsymbol{X}) := [\nabla f_1(\boldsymbol{x}_1), \nabla f_2(\boldsymbol{x}_2), \ldots, \nabla f_n(\boldsymbol{x}_n)] \in \mathbb{R}^{d \times n}$ denote the collection of local gradients.

The average $\bar{\boldsymbol{x}} := \frac{1}{n}\boldsymbol{X}\mathbf{1} \in \mathbb{R}^d$, and $\bar{\boldsymbol{v}} := \frac{1}{n}\nabla F(\boldsymbol{X})\mathbf{1} \in \mathbb{R}^d$.

Plain Gradients vs. Gradient Tracking

Let $\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ denote the collection of parameters from all clients, and $\nabla F(\boldsymbol{X}) := [\nabla f_1(\boldsymbol{x}_1), \nabla f_2(\boldsymbol{x}_2), \ldots, \nabla f_n(\boldsymbol{x}_n)] \in \mathbb{R}^{d\times n}$ denote the collection of local gradients.
The average $\bar{\boldsymbol{x}} := \frac{1}{n}\boldsymbol{X}\mathbf{1} \in \mathbb{R}^d$, and $\bar{\boldsymbol{v}} := \frac{1}{n}\nabla F(\boldsymbol{X})\mathbf{1} \in \mathbb{R}^d$.

- **Issue of plain gradients:** $\boldsymbol{X}^{t+1} = \boldsymbol{X}^t \boldsymbol{W} - \eta \nabla F(\boldsymbol{X}^t)$

Suppose that the model parameters have reached consensus and $\boldsymbol{x}_i^t = \boldsymbol{x}^*$ for all $i \in [n]$. Then the plain gradients will let $\boldsymbol{x}_i^{t+1}$ move away from the solution $\boldsymbol{x}^*$, i.e., $\boldsymbol{x}_i^{t+1} = (\boldsymbol{X}^t \boldsymbol{W})_i - \eta \nabla f_i(\boldsymbol{x}_i^t) = \boldsymbol{x}^* - \eta \nabla f_i(\boldsymbol{x}^*) \neq \boldsymbol{x}^*$.
Note that $\frac{1}{n}\sum_{j=1}^n \nabla f_j(\boldsymbol{x}^*) = 0 \;\; \not\Rightarrow \;\; \nabla f_i(\boldsymbol{x}^*) = 0$

Plain Gradients vs. Gradient Tracking

Let $\boldsymbol{X} := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ denote the collection of parameters from all clients, and $\nabla F(\boldsymbol{X}) := [\nabla f_1(\boldsymbol{x}_1), \nabla f_2(\boldsymbol{x}_2), \ldots, \nabla f_n(\boldsymbol{x}_n)] \in \mathbb{R}^{d\times n}$ denote the collection of local gradients.
The average $\bar{\boldsymbol{x}} := \frac{1}{n}\boldsymbol{X}\mathbf{1} \in \mathbb{R}^d$, and $\bar{\boldsymbol{v}} := \frac{1}{n}\nabla F(\boldsymbol{X})\mathbf{1} \in \mathbb{R}^d$.

- **Issue of plain gradients:** $\boldsymbol{X}^{t+1} = \boldsymbol{X}^t\boldsymbol{W} - \eta\nabla F(\boldsymbol{X}^t)$

Suppose that the model parameters have reached consensus and $\boldsymbol{x}_i^t = \boldsymbol{x}^*$ for all $i \in [n]$. Then the plain gradients will let $\boldsymbol{x}_i^{t+1}$ move away from the solution $\boldsymbol{x}^*$, i.e., $\boldsymbol{x}_i^{t+1} = (\boldsymbol{X}^t\boldsymbol{W})_i - \eta\nabla f_i(\boldsymbol{x}_i^t) = \boldsymbol{x}^* - \eta\nabla f_i(\boldsymbol{x}^*) \neq \boldsymbol{x}^*$.
Note that $\frac{1}{n}\sum_{j=1}^n \nabla f_j(\boldsymbol{x}^*) = 0 \ \not\Rightarrow\ \nabla f_i(\boldsymbol{x}^*) = 0$

- **Benefit of gradient tracking:**

$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t\boldsymbol{W} - \eta\boldsymbol{V}^t; \quad \boldsymbol{V}^{t+1} = \boldsymbol{V}^t\boldsymbol{W} + \nabla F(\boldsymbol{X}^{t+1}) - \nabla F(\boldsymbol{X}^t)$
It gives $\lim_{t\to\infty}\boldsymbol{V}^t = \bar{\boldsymbol{v}}^t\mathbf{1}^\top$, $\boldsymbol{x}_i^{t+1} = (\boldsymbol{X}^t\boldsymbol{W})_i - (\eta\boldsymbol{V}^t)_i = \boldsymbol{x}^* - \eta\bar{\boldsymbol{v}}^* = \boldsymbol{x}^*$

Our BEER Algorithm

Algorithm 1 BEER: BEtter comprEssion for decentRalized optimization

---

Input: Initial point $\boldsymbol{X}^0 = \boldsymbol{x}_0 \mathbf{1}^\top$, $\boldsymbol{G}^0 = \mathbf{0}$, $\boldsymbol{H}^0 = \mathbf{0}$, $\boldsymbol{V}^0 = \nabla F(\boldsymbol{X}_0)$,
step size η, mixing step size γ, minibatch size b

for $t = 0, 1, \ldots$ **do**

EF21

$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t + \gamma \boldsymbol{H}^t(\boldsymbol{W} - \boldsymbol{I}) - \eta \boldsymbol{V}^t$

$\boldsymbol{H}^{t+1} = \boldsymbol{H}^t + \mathcal{C}(\boldsymbol{X}^{t+1} - \boldsymbol{H}^t)$

gradient tracking

$\boldsymbol{V}^{t+1} = \boldsymbol{V}^t + \gamma \boldsymbol{G}^t(\boldsymbol{W} - \boldsymbol{I}) + \tilde{\nabla}_b F(\boldsymbol{X}^{t+1}) - \tilde{\nabla}_b F(\boldsymbol{X}^t)$

$\boldsymbol{G}^{t+1} = \boldsymbol{G}^t + \mathcal{C}(\boldsymbol{V}^{t+1} - \boldsymbol{G}^t)$

EF21

end for

Proof Sketch of BEER

- Compression error: $\Omega_1^t := \mathbb{E}\|\boldsymbol{H}^t - \boldsymbol{X}^t\|_{\mathrm{F}}^2, \quad \Omega_2^t := \mathbb{E}\|\boldsymbol{G}^t - \boldsymbol{V}^t\|_{\mathrm{F}}^2.$
- Consensus error: $\Omega_3^t := \mathbb{E}\|\boldsymbol{X}^t - \bar{\boldsymbol{x}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2, \quad \Omega_4^t := \mathbb{E}\|\boldsymbol{V}^t - \bar{\boldsymbol{v}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2.$

Proof Sketch of BEER

- Compression error: $\Omega_1^t := \mathbb{E}\|\boldsymbol{H}^t - \boldsymbol{X}^t\|_{\mathrm{F}}^2, \quad \Omega_2^t := \mathbb{E}\|\boldsymbol{G}^t - \boldsymbol{V}^t\|_{\mathrm{F}}^2.$
- Consensus error: $\Omega_3^t := \mathbb{E}\|\boldsymbol{X}^t - \bar{\boldsymbol{x}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2, \quad \Omega_4^t := \mathbb{E}\|\boldsymbol{V}^t - \bar{\boldsymbol{v}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2.$
- We prove that $\boldsymbol{\Omega_i^{t+1} \leq (1 - a_i)\Omega_i^t + b_i}, \quad \forall i \in \{1, 2, 3, 4\}.$

Proof Sketch of BEER

- Compression error: $\Omega_1^t := \mathbb{E}\|\boldsymbol{H}^t - \boldsymbol{X}^t\|_{\mathrm{F}}^2, \quad \Omega_2^t := \mathbb{E}\|\boldsymbol{G}^t - \boldsymbol{V}^t\|_{\mathrm{F}}^2.$
- Consensus error: $\Omega_3^t := \mathbb{E}\|\boldsymbol{X}^t - \bar{\boldsymbol{x}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2, \quad \Omega_4^t := \mathbb{E}\|\boldsymbol{V}^t - \bar{\boldsymbol{v}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2.$
- We prove that $\boldsymbol{\Omega_i^{t+1} \leq (1 - a_i)\Omega_i^t + b_i}$, $\forall i \in \{1, 2, 3, 4\}.$
- We define the Lyapunov function:

$$\Phi_t = \mathbb{E}f(\bar{\boldsymbol{x}}^t) - f^* + c_1\Omega_1^t + c_2\Omega_2^t + c_3\Omega_3^t + c_4\Omega_4^t.$$

Proof Sketch of BEER

- Compression error: $\Omega_1^t := \mathbb{E}\|\boldsymbol{H}^t - \boldsymbol{X}^t\|_{\mathrm{F}}^2, \quad \Omega_2^t := \mathbb{E}\|\boldsymbol{G}^t - \boldsymbol{V}^t\|_{\mathrm{F}}^2.$
- Consensus error: $\Omega_3^t := \mathbb{E}\|\boldsymbol{X}^t - \bar{\boldsymbol{x}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2, \quad \Omega_4^t := \mathbb{E}\|\boldsymbol{V}^t - \bar{\boldsymbol{v}}^t\mathbf{1}^\top\|_{\mathrm{F}}^2.$
- We prove that $\boldsymbol{\Omega_i^{t+1} \le (1-a_i)\Omega_i^t + b_i}, \quad \forall i \in \{1,2,3,4\}.$
- We define the Lyapunov function:

$$\Phi_t = \mathbb{E} f(\bar{\boldsymbol{x}}^t) - f^* + c_1\Omega_1^t + c_2\Omega_2^t + c_3\Omega_3^t + c_4\Omega_4^t.$$

- We prove that $\boldsymbol{\Phi_{t+1} \le \Phi_t - \frac{\eta}{2}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2}$ and then obtain the convergence result

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{\boldsymbol{x}}^t)\|^2 \le \frac{2(\Phi_0 - \Phi_T)}{\eta T} = O\left(\frac{1}{T}\right).$$

Conclusion

- We propose a fast compressed algorithm BEER for decentralized nonconvex optimization.

- We show that BEER converges at a faster rate of $O(1/T)$, improving the state-of-the-art rate $O((G/T)^{2/3})$, where T is the number of communication rounds and G measures the data heterogeneity/bounded gradient assumption.

- In sum, BEER removes the strong assumptions (so it can deal with heterogeneous data setting) and also enjoys a faster convergence rate (it matches the rate without communication compression $O(1/T)$).

Thanks!

Zhize Li