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

1. Problem

2. Related Work

3. Our Approaches
   - Compression framework
   - Gradient tracking

4. Conclusion
Optimization Problem

We consider the decentralized optimization problem:

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

\( f(x) \): model parameters,
\( n \): number of clients,
\( f_i(x) \): loss function on client \( i \), \( f_i(x) := \mathbb{E}_{\xi \sim D_i} f(x; \xi) \), where \( D_i \) 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 $D_i$ 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 $C : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an $\alpha$-compression operator if for all $x \in \mathbb{R}^d$, it satisfies

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

(2)

In particular, no compression ($C(x) \equiv x$) 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 $C : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an $\alpha$-compression operator if for all $x \in \mathbb{R}^d$, it satisfies

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

(2)

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

**Examples:** $\text{random}_k(x) = x \odot u$ (where $u$ is a uniformly random binary vector with $k$ nonzero entries, $\odot$ denotes element-wise product) satisfies (2) with $\alpha = k/d$. $\text{top}_k(x)$ also satisfies (2) with $\alpha = k/d$. 
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 \textit{slow convergence rates} (need more communication rounds) and require \textit{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:** $\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \| \nabla f(x; \xi_i) \|^2 \leq G^2$
- **Bounded dissimilarity:** $\mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \|^2 \leq G^2$
## Result Comparison

**Table:** Decentralized nonconvex optimization with communication compression

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

$T$: number of communication rounds  
$n$: total number of clients  
$G$: bounded gradient/dissimilarity assumption  

$$ (\mathbb{E}_{\xi_i \sim D_i} \| \nabla f(x; \xi_i) \|^2 \leq G^2 \text{ or } \mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \|^2 \leq G^2) $$
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 $C$, s.t. $\mathbb{E}[\|C(x) - x\|^2] \leq (1 - \alpha)\|x\|^2$.
- We point out that direct compression framework
  $$x^{t+1} = x^t - \eta \frac{1}{n} \sum_{i=1}^{n} C(\nabla f_i(x^t))$$ does not work.
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. 
\[ \mathbb{E}[\| \mathcal{C}(x) - x \|^2] \leq (1 - \alpha)\|x\|^2. \]

We point out that \textbf{direct compression framework} 
\[ x^{t+1} = x^t - \eta \frac{1}{n} \sum_{i=1}^{n} \mathcal{C}(\nabla f_i(x^t)) \]
\text{does not work.}

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

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

If algorithm starts with \( x^0 = (b, b, b) \), then \( \nabla f_1(x^0) = b(-15, 13, 13)^T \), \( \nabla f_2(x^0) = b(13, -15, 13)^T \), and \( \nabla f_3(x^0) = b(13, 13, -15)^T \).
Direct Compression Framework

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

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

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

If the compressor is \( \text{top}_1 \), we have \( C(\nabla f_1(x^0)) = b(-15, 0, 0)^T \),
\( C(\nabla f_2(x^0)) = b(0, -15, 0)^T \), \( C(\nabla f_3(x^0)) = b(0, 0, -15)^T \),
Direct Compression Framework

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

A counter-example: consider \( n = 3 \) and let \( f_i(x) = (a_i^T x)^2 + \frac{1}{2}\|x\|^2 \), where \( a_1 = (-4, 3, 3)^T \), \( a_2 = (3, -4, 3)^T \) and \( a_3 = (3, 3, -4)^T \).
If algorithm starts with \( x^0 = (b, b, b) \), then \( \nabla f_1(x^0) = b(-15, 13, 13)^T \), \( \nabla f_2(x^0) = b(13, -15, 13)^T \), and \( \nabla f_3(x^0) = b(13, 13, -15)^T \).
If the compressor is top1, we have \( C(\nabla f_1(x^0)) = b(-15, 0, 0)^T \), \( C(\nabla f_2(x^0)) = b(0, -15, 0)^T \), \( C(\nabla f_3(x^0)) = b(0, 0, -15)^T \), and the next iteration \( x^1 = x^0 - \eta \frac{1}{3} \sum_{i=1}^{3} C(\nabla f_i(x^0)) = (1 + 5\eta)x^0 \), and then \( x^t = (1 + 5\eta)^t x^0 \) 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)).

1. Each client $i \in [n]$ sets the zero initial error $e_i^0 = 0$
2. Each client $i \in [n]$ compresses its initial gradient $g_i^0 = C(\gamma \nabla f_i(x^0))$
3. for $t = 0, 1, 2, \ldots$ do
4. Server updates $x^{t+1} = x^t - \frac{1}{n} \sum_{i=1}^{n} g_i^t$
5. for all clients $i = 1, 2, \ldots, n$ do in parallel
6. 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: $g_i^{t+1} = C(e_i^{t+1} + \gamma \nabla f_i(x^{t+1}))$
7. 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: $g^{t+1} = C(e^{t+1} + \gamma \nabla f(x^{t+1}))$

**EF21** (Richtárik et al., 2021)
1. Model update: $x^{t+1} = x^t - \gamma g^t$
2. Update with a shifted compression: $g^{t+1} = g^t + C(\nabla f(x^{t+1}) - g^t)$
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: \( g^{t+1} = C(e^{t+1} + \gamma \nabla f(x^{t+1})) \)

**EF21** (Richtárik et al., 2021)
1. Model update: \( x^{t+1} = x^t - \gamma g^t \)
2. Update with a shifted compression: \( g^{t+1} = g^t + C(\nabla f(x^{t+1}) - g^t) \)

If compressor \( C \) is additive and positively homogeneous, \( \text{EF} = \text{EF21} \).

\[
g^{t+1} = C(e^{t+1} + \gamma \nabla f(x^{t+1})) = C(e^t + \gamma \nabla f(x^t) - g^t + \gamma \nabla f(x^{t+1})) \\
= C(e^t + \gamma \nabla f(x^t)) + C(\gamma \nabla f(x^{t+1}) - g^t) = g^t + C(\gamma \nabla f(x^{t+1}) - g^t).
\]

Let \( g^t \) denote \( \hat{\gamma} \hat{g}^t \), then \( g^{t+1} = \gamma(\hat{g}^t + C(\nabla f(x^t) - \hat{g}^t)) = \gamma \hat{g}^{t+1} \).
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)).
Algorithm 4 CHOCO-SGD (Koloskova et al., 2019) as Error Feedback

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

1: for $t$ in $0 \ldots T - 1$ do \{in parallel for all workers $i \in [n]$\}
2: $x_i^{(t)} := x_i^{(t-1)} + \gamma \sum_{j: \{i,j\} \in E} w_{ij} (\hat{x}_j^{(t)} - \hat{x}_i^{(t)})$
3: $v_i^{(t)} = x_i^{(t)} - x_i^{(t-1)} + m_i^{(t)}$
4: $q_i^{(t)} := Q(v_i^{(t)})$
5: $m_i^{(t+1)} = v_i^{(t)} - q_i^{(t)}$
6: for neighbors $j: \{i,j\} \in E$ (including $\{i\} \in E$) do
7: Send $q_i^{(t)}$ and receive $q_j^{(t)}$
8: $\hat{x}_j^{(t+1)} := q_j^{(t)} + \hat{x}_j^{(t)}$
9: end for
10: Sample $\xi_i^{(t)}$, compute gradient $g_i^{(t)} := \nabla F_i(x_i^{(t)}; \xi_i^{(t)})$
11: $x_i^{(t+\frac{1}{2})} := x_i^{(t)} - \eta g_i^{(t)}$
12: end for

< modified gossip averaging
Error Feedback (EF)
< compression
< memory update
< communication
< local update
plain gradients
< stochastic gradient update
Our BEER Algorithm

Algorithm 1 BEER: BEtter compresSion for decentRalized optimization

1: **Input:** Initial point $X^0 = x_0 1^T$, $G^0 = 0$, $H^0 = 0$, $V^0 = \nabla F(X_0)$, step size $\eta$, mixing step size $\gamma$, minibatch size $b$

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

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

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

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

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

7: **end for**
Plain Gradients vs. Gradient Tracking

Let $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ denote the collection of parameters from all clients, and

$$\nabla F(\mathbf{X}) := [\nabla f_1(\mathbf{x}_1), \nabla f_2(\mathbf{x}_2), \ldots, \nabla f_n(\mathbf{x}_n)] \in \mathbb{R}^{d \times n}$$

denote the collection of local gradients.

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

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

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

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

Suppose that the model parameters have reached consensus and \( \mathbf{x}_i^t = \mathbf{x}^* \) for all \( i \in [n] \). Then the plain gradients will let \( \mathbf{x}_i^{t+1} \) move away from the solution \( \mathbf{x}^* \), i.e.,

\[
\mathbf{x}_i^{t+1} = (\mathbf{X}^{t} \mathbf{W})_i - \eta \nabla f_i(\mathbf{x}_i^t) = \mathbf{x}^* - \eta \nabla f_i(\mathbf{x}^*) \neq \mathbf{x}^*.
\]

Note that

\[
\frac{1}{n} \sum_{j=1}^{n} \nabla f_j(\mathbf{x}^*) = 0 \iff \nabla f_i(\mathbf{x}^*) = 0
\]
Plain Gradients vs. Gradient Tracking

Let $X := [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$ denote the collection of parameters from all clients, and $\nabla F(X) := [\nabla f_1(x_1), \nabla f_2(x_2), \ldots, \nabla f_n(x_n)] \in \mathbb{R}^{d \times n}$ denote the collection of local gradients. The average $\bar{x} := \frac{1}{n}X1 \in \mathbb{R}^d$, and $\bar{v} := \frac{1}{n}\nabla F(X)1 \in \mathbb{R}^d$.

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

Suppose that the model parameters have reached consensus and $x_i^t = x^*$ for all $i \in [n]$. Then the plain gradients will let $x_i^{t+1}$ move away from the solution $x^*$, i.e., $x_i^{t+1} = (X^tW)_i - \eta\nabla f_i(x_i^t) = x^* - \eta\nabla f_i(x^*) \neq x^*$.

Note that $\frac{1}{n} \sum_{j=1}^{n} \nabla f_j(x^*) = 0 \iff \nabla f_i(x^*) = 0$

**Benefit of gradient tracking:**

$X^{t+1} = X^tW - \eta V^t; \quad V^{t+1} = V^tW + \nabla F(X^{t+1}) - \nabla F(X^t)$

It gives $\lim_{t \to \infty} V^t = \bar{v}^t1^\top$, $x_i^{t+1} = (X^tW)_i - (\eta V^t)_i = x^* - \eta\bar{v}^* = x^*$
Our BEER Algorithm

**Algorithm 1 BEER: BEtter comprEssion for decentRalized optimization**

1: **Input:** Initial point $X^0 = x_0 1^T$, $G^0 = 0$, $H^0 = 0$, $V^0 = \nabla F(X_0)$, step size $\eta$, mixing step size $\gamma$, minibatch size $b$

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

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

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

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

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

7: **end for**
Proof Sketch of BEER

- **Compression error:** $\Omega_1^t := \mathbb{E} \| H^t - X^t \|_F^2$, $\Omega_2^t := \mathbb{E} \| G^t - V^t \|_F^2$.
- **Consensus error:** $\Omega_3^t := \mathbb{E} \| X^t - \bar{x}^t 1^\top \|_F^2$, $\Omega_4^t := \mathbb{E} \| V^t - \bar{v}^t 1^\top \|_F^2$. 

We define the Lyapunov function $\mathcal{L}_t := \mathbb{E} f(x_t) + c_1^t + c_2^t + c_3^t + c_4^t$. We prove that $\mathcal{L}_t$ decreases and then obtain the convergence result $\mathcal{T} \mathcal{X}_t = 0 \implies \mathcal{T} \mathcal{F} = O_{1\mathcal{T}}$. 
Proof Sketch of BEER

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

Zhize Li (CMU)
Proof Sketch of BEER

- Compression error: \( \Omega_1^t := \mathbb{E}\|H^t - X^t\|_F^2, \quad \Omega_2^t := \mathbb{E}\|G^t - V^t\|_F^2. \)
- Consensus error: \( \Omega_3^t := \mathbb{E}\|X^t - \bar{x}^t1^\top\|_F^2, \quad \Omega_4^t := \mathbb{E}\|V^t - \bar{v}^t1^\top\|_F^2. \)
- We prove that \( \Omega_i^{t+1} \leq (1 - a_i)\Omega_i^t + b_i, \quad \forall i \in \{1, 2, 3, 4\}. \)
- We define the Lyapunov function:
  \[
  \Phi_t = \mathbb{E}f(\bar{x}^t) - f^* + c_1\Omega_1^t + c_2\Omega_2^t + c_3\Omega_3^t + c_4\Omega_4^t.
  \]
Proof Sketch of BEER

- Compression error: \( \Omega_1^t := \mathbb{E}\|H^t - X^t\|^2_F, \quad \Omega_2^t := \mathbb{E}\|G^t - V^t\|^2_F. \)
- Consensus error: \( \Omega_3^t := \mathbb{E}\|X^t - \bar{x}^t1^\top\|^2_F, \quad \Omega_4^t := \mathbb{E}\|V^t - \bar{v}^t1^\top\|^2_F. \)

We prove that \( \Omega_i^{t+1} \leq (1 - a_i)\Omega_i^t + b_i, \quad \forall i \in \{1, 2, 3, 4\}. \)

We define the Lyapunov function:
\[
\Phi_t = \mathbb{E}f(\bar{x}^t) - f^* + c_1\Omega_1^t + c_2\Omega_2^t + c_3\Omega_3^t + c_4\Omega_4^t.
\]

We prove that \( \Phi_{t+1} \leq \Phi_t - \frac{\eta}{2}\mathbb{E}\|\nabla f(\bar{x}^t)\|^2 \) and then obtain the convergence result
\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\bar{x}^t)\|^2 \leq \frac{2(\Phi_0 - \Phi_T)}{\eta T} = O\left(\frac{1}{T}\right).
\]
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