Coping with Heterogeneity and Privacy in Communication-Efficient Federated Optimization

Yuejie Chi

Carnegie Mellon University

FedVision@CVPR
Jun. 2022
Acknowledgements

Zhize Li
CMU

Boyue Li
CMU

Haoyu Zhao
Princeton

Peter Richtarik
KAUST
Empirical Risk Minimization (ERM)

Given a set of data $\mathcal{M}$,

$$\minimize_{\mathbf{x}} \ f(\mathbf{x}) = \frac{1}{N} \sum_{\mathbf{z} \in \mathcal{M}} \ell(\mathbf{x}; \mathbf{z})$$

Here, $N =$ number of total samples.

- **convex**: least squares, logistic regression
- **non-convex**: PCA, training neural networks (focus of this talk)
Distributed ERM

**Distributed/Federated learning:** due to privacy and scalability, data are distributed at multiple locations / workers / agents.

Let $\mathcal{M} = \bigcup_i \mathcal{M}_i$ be a data partition with equal splitting:

$$f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad \text{where} \quad f_i(x) := \frac{1}{(N/n)} \sum_{z \in \mathcal{M}_i} \ell(x; z).$$

$n = \text{number of agents}$

$N/n = \text{number of local samples}$

$m$
Challenges in federated/decentralized learning

- **Communication efficiency**: limited bandwidth, stragglers, ...

- **Heterogeneity**: non-iid data across the agents

- **Privacy**: does not come for free without sharing data
Communication efficiency

Communication cost = Communication rounds $\times$ Cost per round

• Local method: perform more local computation to reduce communication rounds, e.g. FedAvg (McMahan et al., 2016).

• Communication compression: compress the message into fewer bits, e.g. sparsification or quantization (Alistarh et al., 2017).

We will focus on communication compression methods.
Communication efficiency

Communication cost = Communication rounds $\times$ Cost per round

- **Local method**: perform more local computation to reduce communication rounds, e.g. FedAvg \cite{McMahan2016}.
Communication efficiency

Communication cost = Communication rounds $\times$ Cost per round

- **Local method**: perform more local computation to reduce communication rounds, e.g. FedAvg (McMahan et al., 2016).

- **Communication compression**: compress the message into fewer bits, e.g. sparsification or quantization (Alistarh et al., 2017).
Communication efficiency

Communication cost = Communication rounds × Cost per round

- **Local method**: perform more local computation to reduce communication rounds, e.g. FedAvg (McMahan et al., 2016).

- **Communication compression**: compress the message into fewer bits, e.g. sparsification or quantization (Alistarh et al., 2017).

We will focus on communication compression methods.
Two distributed schemes
Two distributed schemes

Server/client model

PS coordinates *global* information sharing
Two distributed schemes

**Server/client model**
PS coordinates *global* information sharing

**Network/decentralized model**
agents share *local* information over a graph topology
Two distributed schemes

Server/client model
PS coordinates *global* information sharing

Network/decentralized model
agents share *local* information over a graph topology

*Coping with heterogeneity*
Two distributed schemes

Server/client model
PS coordinates *global* information sharing

*Coping with privacy*

Network/decentralized model
agents share *local* information over a graph topology

*Coping with heterogeneity*
A prelude: what should we compress?

What about $x^{t+1} = x^t - \eta \sum_{i=1}^{n} C(\nabla f_i(x^t))$?

Somewhat surprisingly, direct compression doesn't work!
A prelude: what should we compress?

What about

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \frac{1}{n} \sum_{i=1}^{n} C(\nabla f_i(\mathbf{x}^t))$$?
A prelude: what should we compress?

What about

\[ x^{t+1} = x^t - \eta \frac{1}{n} \sum_{i=1}^{n} C(\nabla f_i(x^t)) \]?

Somewhat surprisingly, *direct compression* doesn’t 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 $a_3 = (3, 3, -4)^\top$. 

Zhize Li
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 $a_3 = (3, 3, -4)^\top$.

Let $x^0 = (b, b, b)$, and the compressor be top$_1$,

\[
\nabla f_1(x^0) = b(-15, 13, 13)^\top \quad \rightarrow \quad C(\nabla f_1(x^0)) = b(-15, 0, 0)^\top \\
\nabla f_2(x^0) = b(13, -15, 13)^\top \quad \rightarrow \quad C(\nabla f_2(x^0)) = b(0, -15, 0)^\top \\
\nabla f_3(x^0) = b(13, 13, -15)^\top \quad \rightarrow \quad C(\nabla f_3(x^0)) = b(0, 0, -15)^\top
\]
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 $a_3 = (3, 3, -4)^T$.

- Let $x^0 = (b, b, b)$, and the compressor be top$_1$,
  \[
  \nabla f_1(x^0) = b(-15, 13, 13)^T \quad \rightarrow \quad C(\nabla f_1(x^0)) = b(-15, 0, 0)^T \\
  \nabla f_2(x^0) = b(13, -15, 13)^T \quad \rightarrow \quad C(\nabla f_2(x^0)) = b(0, -15, 0)^T \\
  \nabla f_3(x^0) = b(13, 13, -15)^T \quad \rightarrow \quad C(\nabla f_3(x^0)) = b(0, 0, -15)^T 
  \]

- 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.
A better scheme: shift compression

(Stich et al., 2018; Richtárik et al., 2021)

- Model update:

\[ \mathbf{x}^{t+1} = \mathbf{x}^t - \frac{\eta}{n} \sum_{i=1}^{n} g_i^t \]

— \( g_i^t \) is the compressed surrogate of \( \nabla f_i(\mathbf{x}^t) \)
A better scheme: shift compression

(Stich et al., 2018; Richtárik et al., 2021)

- Model update:

\[ \mathbf{x}^{t+1} = \mathbf{x}^t - \eta \sum_{i=1}^{n} \mathbf{g}^t_i \]

— \( \mathbf{g}^t_i \) is the compressed surrogate of \( \nabla f_i(\mathbf{x}^t) \)

- Update \( \mathbf{g}^t_i \) with a shift compression:

\[ \mathbf{g}_i^{t+1} = \mathbf{g}_i^t + C(\nabla f_i(\mathbf{x}^{t+1}) - \mathbf{g}_i^t) \]

— \( \mathbf{g}_i^t \) is constructed accumulatively over time
A better scheme: shift compression

(Stich et al., 2018; Richtárik et al., 2021)

• Model update:

\[ \mathbf{x}^{t+1} = \mathbf{x}^t - \eta \sum_{i=1}^{n} g_i^t \]

— \( g_i^t \) is the compressed surrogate of \( \nabla f_i(\mathbf{x}^t) \)

• Update \( g_i^t \) with a shift compression:

\[ g_i^{t+1} = g_i^t + C(\nabla f_i(\mathbf{x}^{t+1}) - g_i^t) \]

— \( g_i^t \) is constructed accumulatively over time

We’ll consider algorithms using shift compression!
BEER: Fast Decentralized Nonconvex Optimization with Communication Compression

Haoyu Zhao
Princeton

Boyue Li
CMU

Zhize Li
CMU

Peter Richtarik
KAUST
Prior art

CHOCO-SGD (Koloskova et al., 2019) / DeepSqueeze (Tang et al., 2019):
Prior art

CHOCO-SGD (Koloskova et al., 2019) / DeepSqueeze (Tang et al., 2019):

- slow convergence rates (need more communication rounds) and
- Incompatible with heterogeneity: bounded gradient or dissimilarity

\[ \mathbb{E}_{\xi_i \sim D_i} \| \nabla f(x; \xi_i) \| \leq G^2 \quad \text{or} \quad \mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \| \leq G^2 \]
Prior art

CHOCO-SGD (Koloskova et al., 2019) / DeepSqueeze (Tang et al., 2019):

- slow convergence rates (need more communication rounds) and
- Incompatible with heterogeneity: bounded gradient or dissimilarity

\[
\mathbb{E}_{\xi_i \sim D_i} \| \nabla f(x; \xi_i) \| \leq G^2 \quad \text{or} \quad \mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \| \leq G^2
\]

Can we converge at the rate \( O \left( \frac{1}{\varepsilon} \right) \) under arbitrary heterogeneity?
Prior art

CHOCO-SGD (Koloskova et al., 2019) / DeepSqueeze (Tang et al., 2019):
- slow convergence rates (need more communication rounds) and
- Incompatible with heterogeneity: bounded gradient or dissimilarity

\[
\mathbb{E}_{\xi_i \sim D_i} \| \nabla f(x; \xi_i) \| \leq G^2 \quad \text{or} \quad \mathbb{E}_i \| \nabla f_i(x) - \nabla f(x) \| \leq G^2
\]

Can we converge at the rate \( O\left(\frac{1}{\varepsilon}\right) \) under arbitrary heterogeneity?

Yes, by using gradient tracking!
Decentralized gradient descent: a naive extension

**Centralized Gradient Descent (GD):**

\[
x^t = x^{t-1} - \eta \nabla f(x^{t-1})
\]

Constant step size, linear convergence for strongly convex problems.

**Decentralized Gradient Descent (DGD):**

\[
x^t_i = \sum_j w_{ij} x^t_{i-1} - \eta \nabla f_i(x^t_i)
\]

Constant step size, does not converge!

At optimal point \(x^\star\):

\(\nabla f(x^\star) = 0\), but \(\nabla f_i(x^\star) \neq 0\)

How do we fix this?
Decentralized gradient descent: a naive extension

Centralized Gradient Descent (GD):

\[ x^t = x^{t-1} - \eta \nabla f(x^{t-1}) \]

Decentralized Gradient Descent (DGD):

\[ x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta \nabla f_i(x_i^{t-1}) \]

- mixing
- local gradient
Decentralized gradient descent: a naive extension

Centralized Gradient Descent (GD):

\[ x^t = x^{t-1} - \eta \nabla f(x^{t-1}) \]

Constant step size, linear convergence for strongly convex problems.

Decentralized Gradient Descent (DGD):

\[ x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta \nabla f_i(x_i^{t-1}) \]

\text{mixing} \quad \text{local gradient}
Decentralized gradient descent: a naive extension

Centralized Gradient Descent (GD):

\[ x^t = x^{t-1} - \eta \nabla f(x^{t-1}) \]

Constant step size, linear convergence for strongly convex problems.

Decentralized Gradient Descent (DGD):

\[ x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta \nabla f_i(x_i^{t-1}) \]

Constant step size, does not converge!
Decentralized gradient descent: a naive extension

**Centralized Gradient Descent (GD):**

\[ x^t = x^{t-1} - \eta \nabla f(x^{t-1}) \]

*Constant* step size, *linear* convergence for strongly convex problems.

**Decentralized Gradient Descent (DGD):**

\[ x^t_i = \sum_j w_{ij} x^{t-1}_j - \eta \nabla f_i(x^{t-1}_i) \]

*Constant* step size, *does not* converge!

At optimal point \( x^* \): \( \nabla f(x^*) = 0 \), but \( \nabla f_i(x^*) \neq 0 \)

*How do we fix this?*
Use dynamic average consensus (Zhu and Martinez, 2010) to track the global gradient $s_i^t$:

$$x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta s_i^t$$

$$(s_i^t = \sum_j w_{ij} s_j^{t-1} + \nabla f_i(x_i^t) - \nabla f_i(x_i^{t-1}))$$
DGD with gradient tracking

Use dynamic average consensus ([Zhu and Martinez, 2010](#)) to track the global gradient $s^t_i$:

\[
\begin{align*}
    x^t_i &= \sum_j w_{ij} x^{t-1}_j - \eta s^t_i \\
    s^t_i &= \sum_j w_{ij} s^{t-1}_i + \nabla f_i(x^t_i) - \nabla f_i(x^{t-1}_i)
\end{align*}
\]

This trick, and other alternatives, have been used extensively to fix the non-convergence issue in decentralized optimization.
DGD with gradient tracking

Use dynamic average consensus ([Zhu and Martinez, 2010](#)) to track the global gradient $s_i^t$:

$$x_i^t = \sum_j w_{ij} x_j^{t-1} - \eta s_i^t$$

$$(s_i^t = \sum_j w_{ij} s_j^{t-1} + \nabla f_i(x_i^t) - \nabla f_i(x_i^{t-1})$$

This trick, and other alternatives, have been used extensively to fix the non-convergence issue in decentralized optimization.

- EXTRA ([Shi, Ling, Wu and Yin, 2015](#)); NEXT ([Di Lorenzo and Scutari, 2016](#)); NIDS ([Li, Shi, Yan, 2017](#)); ADD-OPT ([Xi, Xin, and Khan, 2017](#)); DIGING ([Nedic, Olshevsky, and Shi, 2017](#)); DGD ([Qu and Li, 2018](#));
- many, many more...
BEER: gradient tracking + shift compression

\[ X = [x_1, x_2, \cdots, x_n]: \text{local models.} \]
\[ \nabla F(X) = [\nabla f_1(x_1), \nabla f_2(x_2), \cdots, \nabla f_n(x_n)]: \text{local gradients.} \]
BEER: gradient tracking + shift compression

\[ X = [x_1, x_2, \cdots, x_n]: \text{local models.} \]
\[ \nabla F(X) = [\nabla f_1(x_1), \nabla f_2(x_2), \cdots, \nabla f_n(x_n)]: \text{local gradients.} \]

- **model update:**

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

where \( H^t \) is the accumulated compressed surrogate of \( X^t \), and \( V^t \) is the global gradient estimates across the agents.
BEER: gradient tracking + shift compression

\[ X = [x_1, x_2, \ldots, x_n]: \text{local models.} \]
\[ \nabla F(X) = [\nabla f_1(x_1), \nabla f_2(x_2), \ldots, \nabla f_n(x_n)]: \text{local gradients.} \]

- **model update:**

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

where \( H^t \) is the accumulated compressed surrogate of \( X^t \), and \( V^t \) is the global gradient estimates across the agents.

- **gradient tracking:**

\[ V^{t+1} = V^t + \gamma G^t(W - I) + \nabla F(X^{t+1}) - \nabla F(X^t), \]

where \( G^t \) is the accumulated compressed surrogate of \( V^t \).
BEER: gradient tracking + shift compression

\[ X = [x_1, x_2, \cdots, x_n] \] local models.
\[ \nabla F(X) = [\nabla f_1(x_1), \nabla f_2(x_2), \cdots, \nabla f_n(x_n)] \] local gradients.

- **model update:**

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

where \( H^t \) is the accumulated compressed surrogate of \( X^t \), and \( V^t \) is the global gradient estimates across the agents.

- **gradient tracking:**

\[
V^{t+1} = V^t + \gamma G^t(W - I) + \nabla F(X^{t+1}) - \nabla F(X^t),
\]

where \( G^t \) is the accumulated compressed surrogate of \( V^t \).

- Both \( H^t \) and \( G^t \) are updated using **shift compression**.
Theoretical convergence of BEER

**Theorem (Zhao et al., 2022)**

To achieve $\mathbb{E}\|\nabla f(x^\text{output})\|^2 \leq \varepsilon$, BEER requires at most

$$O\left(\frac{1}{\rho^3 \alpha \varepsilon}\right)$$

communication rounds, without the bounded gradient assumption. Here, $\alpha$ is the compression ratio, $\beta$ is the spectral gap of the network.
Theoretical convergence of BEER

**Theorem (Zhao et al., 2022)**

To achieve $\mathbb{E}\|\nabla f(\mathbf{x}^{\text{output}})\|^2 \leq \varepsilon$, BEER requires at most

$$O\left(\frac{1}{\rho^3 \alpha \varepsilon}\right)$$

communication rounds, without the bounded gradient assumption. Here, $\alpha$ is the compression ratio, $\beta$ is the spectral gap of the network.

BEER converges at the rate $O\left(\frac{1}{\varepsilon}\right)$ under arbitrary heterogeneity!
**BEER vs CHOCO-SGD**

**Figure:** Training gradient norm and testing accuracy against communication rounds for classification on the *unshuffled* MNIST dataset using a simple neural network. Both BEER and CHOCO-SGD employ the biased gsgd$_b$ compression with $b = 20$. 
SoteriaFL: A Unified Framework for Private FL with Communication Compression

Zhize Li
CMU

Haoyu Zhao
Princeton

Boyue Li
CMU
Motivation: a unified framework?

- **Privacy**: need to preserve the privacy of local data
- **Communication**: shift compression with many options, e.g. sparsification or quantization
- **Computation**: stochastic local gradient estimators with many options, e.g. SGD, SVRG or SAGA
Motivation: a unified framework?

- **Privacy:** need to preserve the privacy of local data
- **Communication:** shift compression with many options, e.g. sparsification or quantization
- **Computation:** stochastic local gradient estimators with many options, e.g. SGD, SVRG or SAGA

Can we develop a unified framework for private FL with compression, with a characterization of the privacy-utility-communication trade-off?
SoteriaFL: a unified framework for compressed private FL

Highlights of SoteriaFL:

- Flexible local gradient estimators
- Protect local data privacy
- State-of-the-art shift compression scheme
- Privacy-utility-communication trade-offs
SoteriaFL: a unified framework for compressed private FL

Highlights of SoteriaFL:
- Flexible local gradient estimators
- Protect local data privacy
- State-of-the-art shift compression scheme
- Privacy-utility-communication trade-offs

At each client:
- Local gradient estimator
- Gaussian mechanism
- Shift compression
- Shift update

At the server:
- Aggregation
- Model update
- Global shift update
Privacy-utility-communication trade-off

Under $(\epsilon, \delta)$ local differential privacy:

- Utility/accuracy: $\frac{\sqrt{\alpha \log(1/\delta)}}{\epsilon}$
- Communication: $\frac{\epsilon}{\sqrt{\alpha^3 \log(1/\delta)}}$
Provably efficient communication-compressed FL algorithms for heterogeneous and private data!

Future work:
- privacy-preserving decentralized algorithms under data heterogeneity.
1. **BEER**: Fast $O(1/T)$ Rate for Decentralized Nonconvex Optimization with Communication Compression

2. **SoteriaFL**: A Unified Framework for Private Federated Learning with Communication Compression
   Z. Li, H. Zhao, B. Li, and Y. Chi, arXiv today.