Communication Compression for Byzantine Robust Learning: New Efficient Algorithms and Improved Rates

Ahmad Rammal [1, 2] Kaja Gruntkowska [1] Nikita Fedin [3] Eduard Gorbunov [4] Peter Richtárik [1]

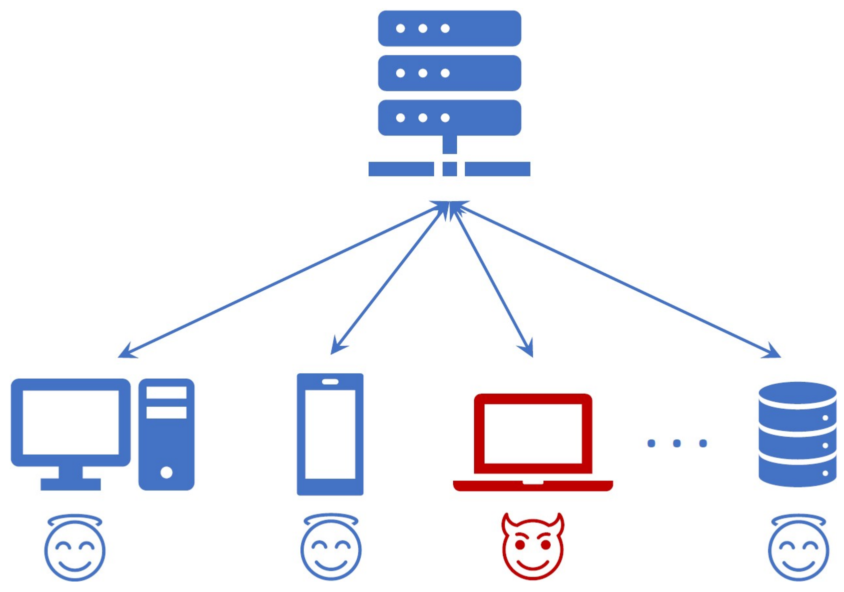
[1]KAUST [2]École Polytechnique [3]MIPT [4]MBZUAI

The Problem

Nonconvex *distributed* optimization problem:

$$\min_{x\in\mathbb{R}^d}\left\{f(x)=\frac{1}{G}\sum_{i\in\mathcal{G}}f_i(x)\right\}$$

$$f_i(x)=\frac{1}{m}\sum_{j=1}^{m}f_{i,j}(x)$$



- $\mathcal{G}$ is the set of *good/regular/non-Byzantine* clients, $|\mathcal{G}|=G$
- $\mathcal{B}$ is the set of *bad/malicious/Byzantine* clients
- $\mathcal{G}\sqcup\mathcal{B}=[n]$, where n is the total number of clients
- $f_i(x)$ – loss of the model x on the data stored on worker i
- $f_{i,j}(x)$ – loss on the j-th example from the local dataset of worker i

Goal: find $\hat{x}$ such that $\mathbb{E}[\|\nabla f(\hat{x})\|^2]\le\varepsilon^2$

Compressed learning

Unbiased compressor

$$\mathbb{E}[\mathcal{Q}(x)]=x,\ \mathbb{E}\left[\|\mathcal{Q}(x)\|^2\right]\le\omega\|x\|^2$$

Example: **Rand-K** $\in\mathbb{U}(d/K)$:

$$\mathcal{Q}(x):=\frac{d}{K}\sum_{i\in S}x_ie_i$$

Contractive compressor

$$\mathbb{E}\left[\|\mathcal{C}(x)-x\|^2\right]\le(1-\alpha)\|x\|^2$$

Example: **Top-K** $\in\mathbb{B}(K/d)$:

$$\mathcal{C}(x):=\sum_{i=d-K+1}^{d}x_{(i)}e_{(i)}$$

Assumptions

L-smoothness: The function $f:\mathbb{R}^d\to\mathbb{R}$ is L-smooth, i.e.,

$$\|\nabla f(x)-\nabla f(y)\|\le L\|x-y\|$$

for any $x,y\in\mathbb{R}^d$. Moreover, $f_*=\inf_{x\in\mathbb{R}^d}f(x)>-\infty$.

Global Hessian variance [1]: There exists $L_\pm\ge0$ such that for all $x,y\in\mathbb{R}^d$

$$\frac{1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)-\nabla f_i(y)\|^2-\|\nabla f(x)-\nabla f(y)\|^2\le L_\pm^2\|x-y\|^2.$$

Local Hessian variance [2]: There exists $\mathcal{L}_\pm\ge0$ such that for all $x,y\in\mathbb{R}^d$ the unbiased mini-batched estimator $\widehat{\Delta}_i(x,y)$ of $\Delta_i(x,y)=\nabla f_i(x)-\nabla f_i(y)$ with batch size b satisfies

$$\frac{1}{G}\sum_{i\in\mathcal{G}}\mathbb{E}\left[\|\widehat{\Delta}_i(x,y)-\Delta_i(x,y)\|^2\right]\le\frac{\mathcal{L}_\pm^2}{b}\|x-y\|^2.$$

(B,ζ^2)-heterogeneity [2]: There exist $B,\zeta\ge0$ such that for all $x\in\mathbb{R}^d$

$$\frac{1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)-\nabla f(x)\|^2\le B\|\nabla f(x)\|^2+\zeta^2.$$

Main contributions

◇ **Improved complexity bounds:** Two new Byzantine-robust methods with *unbiased* compression: Byz-VR-MARINA 2.0 and Byz-DASHA-PAGE, outperforming the previous SOTA Byz-VR-MARINA by factors of $\sqrt{\max\{\omega,m/b\}}$ and $\sqrt{\max\{\omega^3,m^2\omega/b^2\}}$ in the leading term.

◇ **Smaller size of the neighborhood:** Byz-VR-MARINA 2.0 and Byz-DASHA-PAGE converge to a smaller neighborhood of the solution than their competitors. When $B=0$, we prove that $\mathbb{E}[\|\nabla f(x)\|^2]=\mathcal{O}(c\delta)$, matching the lower bound [3] and improving on $\mathbb{E}[\|\nabla f(x)\|^2]=\mathcal{O}(c\delta/p)$ of Byz-VR-MARINA.

◇ **Higher tolerance to Byzantine workers:** When $B>0$, our results guarantee convergence in the presence of $1/p$ times more Byzantine workers than in the case of Byz-VR-MARINA.

◇ **The first Byzantine-robust methods with EF:** Two new Byzantine-robust methods employing any contractive compressors – Byz-EF21 and Byz-EF21-BC. Additionally, Byz-EF21-BC is the first provably Byzantine-robust algorithm using bidirectional compression.

Table: Summary of the complexity bounds in the general non-convex case. Columns: "Rounds" = the number of communication rounds required to find x such that $\mathbb{E}[\|\nabla f(x)\|^2]\le\varepsilon^2$; "$\varepsilon\le$" = the lower bound for the best achievable accuracy ε; "$\delta<$" = the maximal ratio of Byzantine workers that the method can provably tolerate.

Method	Rounds	$\varepsilon\le$	$\delta<$
Byz-VR-MARINA (1) [2]	$\frac{1}{\varepsilon^2}\left(1+\sqrt{\max\{\omega^2,\frac{m\omega}{b}\}}\left(\sqrt{\frac{1}{G}}+\sqrt{c\delta\max\{\omega,\frac{m}{b}\}}\right)\right)$	$\frac{c\delta\zeta^2}{p-c\delta B}$	$\frac{p}{cB}$
Byz-VR-MARINA 2.0 (1)	$\frac{1}{\varepsilon^2}\left(1+\sqrt{\max\{\omega^2,\frac{m\omega}{b}\}}\left(\sqrt{\frac{1}{G}}+\sqrt{c\delta}\right)\right)$	$\frac{c\delta\zeta^2}{1-c\delta B}$	$\frac{1}{(c+\sqrt{c})B}$
Byz-DASHA-PAGE (1)	$\frac{1}{\varepsilon^2}\left(1+\left(\omega+\frac{\sqrt{m}}{b}\right)\left(\sqrt{\frac{1}{G}}+\sqrt{c\delta}\right)\right)$	$\frac{c\delta\zeta^2}{1-c\delta B}$	$\frac{1}{(c+\sqrt{c})B}$
Byz-EF21 (2)	$\frac{1+\sqrt{c\delta}}{\alpha_D\varepsilon^2}$	$\frac{(c\delta+\sqrt{c\delta})\zeta^2}{1-B(c\delta+\sqrt{c\delta})}$	$\frac{1}{c(B+B^2)}$
Byz-EF21-BC (2)	$\frac{1+\sqrt{c\delta}}{\alpha_D\alpha_P\varepsilon^2}$	$\frac{(c\delta+\sqrt{c\delta})\zeta^2}{1-B(c\delta+\sqrt{c\delta})}$	$\frac{1}{c(B+B^2)}$

(1) For Byz-VR-MARINA (2.0), $p=\min\{1/\omega,b/m\}$; for Byz-DASHA-PAGE $p=b/m$.
(2) These methods use (biased) contractive compression and compute full gradients on regular workers.

Algorithms

Byz-VR-MARINA 2.0

```
Input: starting point x_0 ∈ R^d, stepsize γ > 0, probability p ∈ (0,1], number of iterations T ≥ 1, unbiased compressors {Q_i}_{i∈G}
for t = 0,1,...,T−1 do
    Sample c^{t+1} ~ Bernoulli(p)
    Broadcast g^t to all nodes
    for i ∈ G in parallel do
        x^{t+1} = x^t − γ g^t
        if c^{t+1} = 1 then
            g_i^{t+1} = ∇f_i(x^{t+1})
            Send ∇f_i(x^{t+1}) to the server.
        else
            m_i^{t+1} = Q_i(Δ̂_i(x^{t+1}, x^t))
            g_i^{t+1} = g_i^t + m_i^{t+1}
            Send m_i^{t+1} to the server
        end if
    end for
    g^{t+1} = ARAgg(g_1^{t+1}, ..., g_n^{t+1})
end for
```

Byz-DASHA-PAGE

```
Input: starting point x^0 ∈ R^d, stepsize γ > 0, momentum a ∈ (0,1], probability p ∈ (0,1], number of iterations T ≥ 1, unbiased compressors {Q_i}_{i∈G}
for t = 0,1,...,T−1 do
    Sample c^{t+1} ~ Bernoulli(p)
    Broadcast g^t to all nodes
    for i ∈ G in parallel do
        x^{t+1} = x^t − γ g^t
        if c^{t+1} = 1 then
            h_i^{t+1} = ∇f_i(x^{t+1})
        else
            h_i^{t+1} = h_i^t + Δ̂_i(x^{t+1}, x^t)
        end if
        m_i^{t+1} = Q_i(h_i^{t+1} − h_i^t − a(g_i^t − h_i^t))
        g_i^{t+1} = g_i^t + m_i^{t+1}
        Send m_i^{t+1} to the server
    end for
    g^{t+1} = ARAgg(g_1^{t+1}, ..., g_n^{t+1})
end for
```

Byz-EF21

```
Input: starting point x^0 ∈ R^d, stepsize γ > 0, number of iterations T ≥ 1, biased compressors {C_i}_{i∈G}
for t = 0,1,...,T−1 do
    x^{t+1} = x^t − γ g^t

    Broadcast x^{t+1} to all workers
    for i ∈ G in parallel do

        c_i^t = C_i(∇f_i(x^{t+1}) − g_i^t)
        g_i^{t+1} = g_i^t + c_i^t
        Send message c_i^t to the server
    end for
    g^{t+1} = ARAgg(g_1^{t+1}, ..., g_n^{t+1})
end for
```

Byz-EF21-BC

```
Input: starting point x^0 ∈ R^d, stepsize γ > 0, number of iterations T ≥ 1, biased compressors {C_i^D}_{i∈G}, C^P
for t = 0,1,...,T−1 do
    x^{t+1} = x^t − γ g^t
    s^{t+1} = C^P(x^{t+1} − w^t)
    w^{t+1} = w^t + s^{t+1}
    Broadcast s^{t+1} to all workers
    for i ∈ G in parallel do
        w^{t+1} = w^t + s^{t+1}
        c_i^t = C_i^D(∇f_i(w^{t+1}) − g_i^t)
        g_i^{t+1} = g_i^t + c_i^t
        Send message c_i^t to the server
    end for
    g^{t+1} = ARAgg(g_1^{t+1}, ..., g_n^{t+1})
end for
```

Robust Aggregation

(δ,c)-Robust Aggregator [2]

Assume that $\{x_1,\ldots,x_n\}$ is such that there exists a subset $\mathcal{G}\subseteq[n]$ of size $|\mathcal{G}|=G\ge(1-\delta)n$ with $\delta<0.5$, and $\sigma\ge0$ such that $\frac{1}{G(G-1)}\sum_{i,l\in\mathcal{G}}\mathbb{E}[\|x_i-x_l\|^2]\le\sigma^2$. Then $\widehat{x}$ is a (δ,c)**-Robust Aggregator** ($\widehat{x}=\mathtt{RAgg}(x_1,\ldots,x_n)$) if

$$\mathbb{E}\left[\|\widehat{x}-\overline{x}\|^2\right]\le c\delta\sigma^2$$

for some $c>0$, where $\overline{x}=\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}x_i$. If additionally $\widehat{x}$ is computed without the knowledge of σ^2, then $\widehat{x}$ is a (δ,c)**-Agnostic Robust Aggregator** ((δ,c)-$\mathtt{ARAgg}$) ($\widehat{x}=\mathtt{ARAgg}(x_1,\ldots,x_n)$).

Examples:

$$[\mathtt{CM}(x_1,\ldots,x_n)]_j:=\mathtt{Median}([x_1]_j,\ldots,[x_n]_j)$$

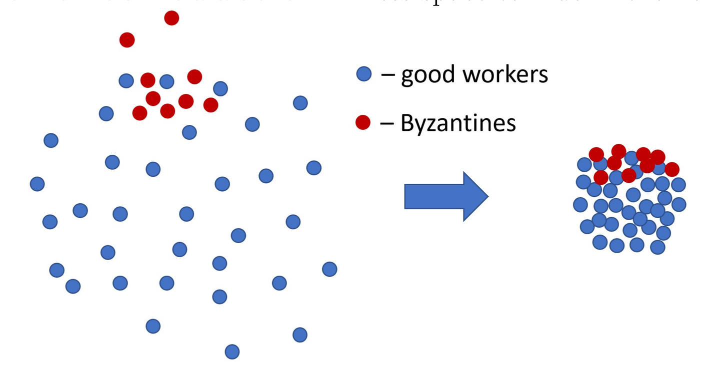
$$\mathtt{GM}(x_1,\ldots,x_n):=\arg\min_{x\in\mathbb{R}^d}\sum_{i=1}^{n}\|x-x_i\|$$

$$\mathtt{Krum}(x_1,\ldots,x_n):=\arg\min_{x_i\in\{x_1,\ldots,x_n\}}\sum_{j\in S_i}\|x_j-x_i\|^2$$

+ Bucketing [3]

1: **Input:** $\{x_1,\ldots,x_n\}$, bucket size $s\in\mathbb{N}$, aggregation rule $\mathtt{Aggr}$
2: Sample a random permutation $\pi=(\pi(1),\ldots,\pi(n))$ of $[n]$
3: Set $y_i=\frac{1}{s}\sum_{k=s(i-1)+1}^{\min\{si,n\}}x_{\pi(k)}$ for $i=1,\ldots,\lceil n/s\rceil$
4: **Return:** $\hat{x}=\mathtt{Aggr}(y_1,\ldots,y_{\lceil n/s\rceil})$

Variance reduction = less space to hide in the noise



Experiments

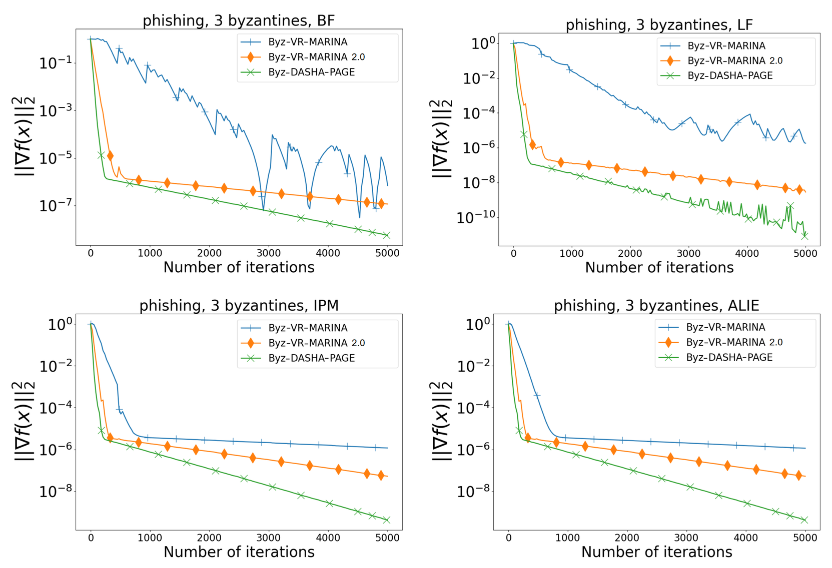


Figure: Logistic regression problem with non-convex regularizer in the homogeneous setting.

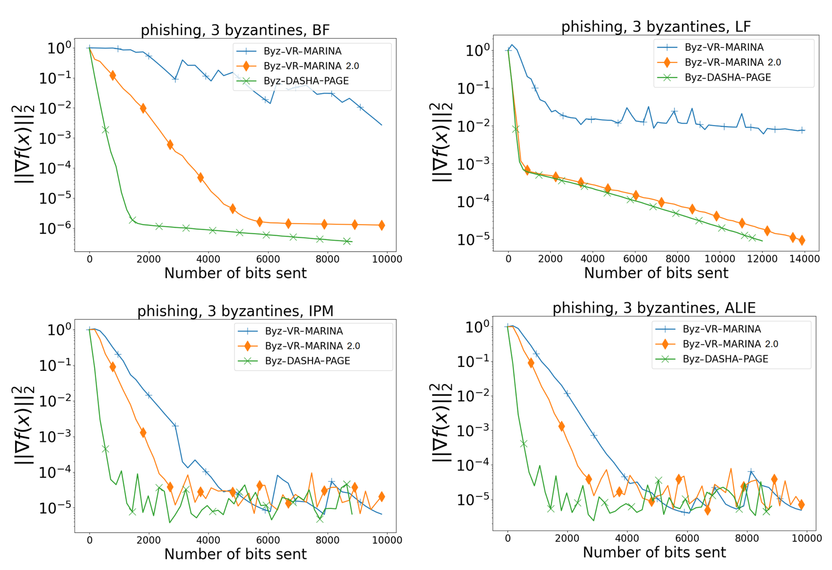


Figure: Logistic regression problem with non-convex regularizer in the heterogeneous setting.

◇ **Bit Flipping (BF):** flip the sign of the updates.
◇ **Label Flipping (LF):** change labels: $y_{i,j}\mapsto-y_{i,j}$.
◇ **A Little Is Enough (ALIE):** estimate the mean $\mu_\mathcal{G}$ and standard deviation $\sigma_\mathcal{G}$ of the regular updates and send $\mu_\mathcal{G}-z\sigma_\mathcal{G}$.
◇ **Inner Product Manipulation (IPM):** send $-\frac{z}{G}\sum_{i\in\mathcal{G}}\nabla f_i(x)$.

References

[1] R. Szlendak, A. Tyurin, and P. Richtárik. Permutation compressors for provably faster distributed nonconvex optimization. *arXiv preprint arXiv:2110.03300*, 2021.

[2] E. Gorbunov, S. Horváth, P. Richtárik, and G. Gidel. Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. *International Conference on Learning Representations*, 2023.

[3] S. P. Karimireddy, L. He, and M. Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. *International Conference on Learning Representations*, 2022.