Byzantine Robustness and Partial Participation Can Be Achieved Simultaneously: Just Clip Gradient Differences

Grigory Malinovsky [1] Peter Richtárik [1] Samuel Horváth [2] Eduard Gorbunov [2]

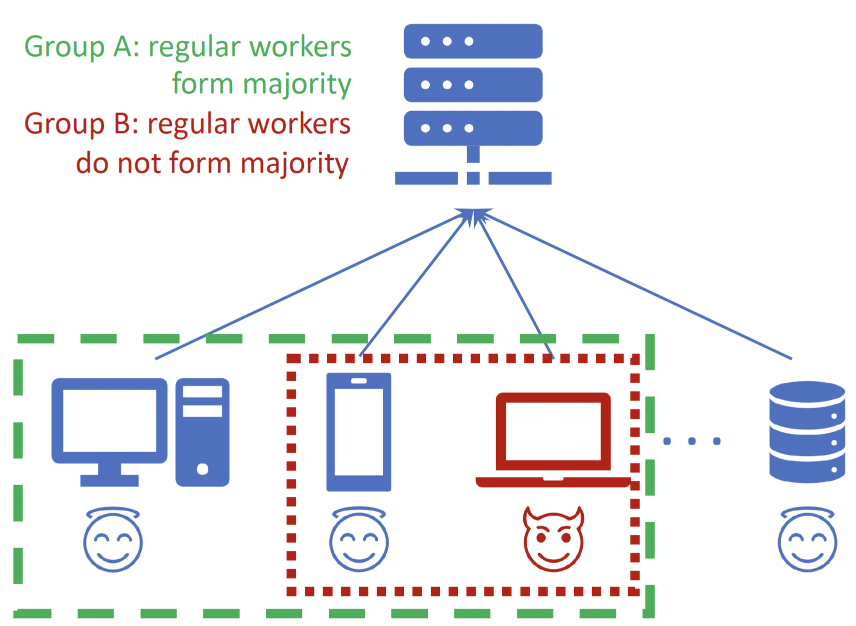
[1]King Abdullah University of Science and Technology [2]Mohamed bin Zayed University of Artificial Intelligence

1. Byzantine-Robust Optimization

Distributed optimization problem:

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

- $\mathcal{G}$ is the set of regular clients
- $\mathcal{B}$ is the set of *Byzantine workers* – the workers that can arbitrarily deviate from the prescribed protocol (maliciously or not) and are assumed to be omniscient
- $\mathcal{G}\sqcup\mathcal{B}=[n]$ is the set of clients participating in training



Main difficulties in Byzantine-robust optimization:

- When functions are arbitrarily heterogeneous, the problem is impossible to solve
- Fraction of Byzantines $\delta={}^B/_n$ should be smaller than $^1/_2$
- Standard approaches based on averaging are vulnerable
- Robust aggregation alone does not ensure robustness [1]
- Non-triviality of partial participation: *all existing approaches are vulnerable* to the situations when regular workers **do not form a majority** during some rounds

2. Robust Aggregation

Popular aggregation rules:

- $\texttt{Krum}(x_1,\ldots,x_n):=\operatorname{argmin}_{x_i\in\{x_1,\ldots,x_n\}}\sum_{j\in S_i}\|x_j-x_i\|^2$ [7], where $S_i\subseteq\{x_1,\ldots,x_n\}$ are $n-|\mathcal{B}|-2$ closest vectors to x_i
- Robust Fed. Averaging: $\texttt{RFA}(x_1,\ldots,x_n):=\operatorname{argmin}_{x\in\mathbb{R}^d}\sum_{i=1}^n\|x-x_i\|$
- Coordinate-wise Median: $[\texttt{CM}(x_1,\ldots,x_n)]_t:=\operatorname{argmin}_{u\in\mathbb{R}}\sum_{i=1}^n|u-[x_i]_t|$

These defenses are vulnerable to Byzantine attacks [8,9] and do not satisfy the following definition.

Definition 1: (δ,c)-Robust Aggregator (modification of the definition from [1])

The quantity $\widehat{x}$ is (δ,c)-**Robust Aggregator** ((δ,c)-RAgg) if

$$\mathbb{E}\left[\|\widehat{x}-\overline{x}\|^2\right]\le c\delta\sigma^2,\quad\text{where}\tag{1}$$

- Input: $\{x_1,x_2,\ldots,x_n\}$
- There exists a subset $\mathcal{G}\subseteq[n]$ of size $|\mathcal{G}|=G\ge(1-\delta)n$ for $\delta<0.5$ such that $\frac{1}{G(G-1)}\sum_{i,l\in\mathcal{G}}\mathbb{E}[\|x_i-x_l\|^2]\le\sigma^2$
- $\overline{x}=\frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}}x_i$
- $\widehat{x}$ is agnostic ((δ,c)-ARAgg), if it can be computed without knowledge of σ

One can robustify $\texttt{Krum}$, $\texttt{RFA}$, and $\texttt{CM}$ using bucketing [1].

Algorithm $\texttt{Bucketing}$: Robust Aggregation using bucketing [1]

Input: $\{x_1,\ldots,x_n\}$, $s\in\mathbb{N}$ – bucket size, Aggr – aggregator
Sample random permutation $\pi=(\pi(1),\ldots,\pi(n))$ of $[n]$
Compute $y_i=\frac{1}{s}\sum_{k=s(i-1)+1}^{\min\{si,n\}}x_{\pi(k)}$ for $i=1,\ldots,\lceil n/s\rceil$
Return: $\widehat{x}=\text{Aggr}(y_1,\ldots,y_{\lceil n/s\rceil})$

Main Contributions

- ◇ **New method: Byz-VR-MARINA-PP.** We develop Byzantine-tolerant Variance-Reduced MARINA with Partial Participation (Byz-VR-MARINA-PP) – **the first distributed method having Byzantine robustness and allowing partial participation of clients**.
- ◇ **New convergence rates.** We derive convergence guarantees for the proposed method under mild assumptions.
- ◇ **New application of gradient clipping.** The key tool that allows our method to withstand Byzantines attacks even when all sampled clients are Byzantine is clipping.

3. Ingredient 1: Variance Reduction

SGD: $x^{k+1}=x^k-\gamma g^k,\quad g^k=\frac{1}{n}\sum_{i=1}^n\nabla f_{i,j_i^k}(x^k)$

✗ Variances of the estimators $\nabla f_{i,j_i^k}(x^k)$ do not go to zero
✗ Byzantines can easily hide in the noise and create a large bias (even if the aggregation is robust)

SAGA [2]: $x^{k+1}=x^k-\gamma g^k,\quad g^k=\frac{1}{n}\sum_{i=1}^n g_i^k$,

$g_i^k=\nabla f_{j_i^k}(x^k)-\nabla f_{i,j_i^k}(w_{i,j_i^k}^k)+\frac{1}{m}\sum_{j=1}^m\nabla f_{i,j}(w_{i,j}^k)$

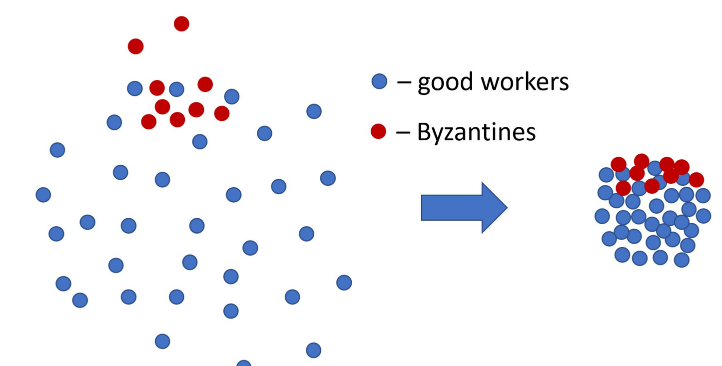
✓ Variances of the estimators g_i^k go to zero
✗ Analysis relies on the unbiasedness: $\mathbb{E}[g_i^k\mid x^k]=\nabla f_i(x^k)$

SARAH/Geom-SARAH/PAGE [3,4,5]:
$x^{k+1}=x^k-\gamma g^k,\quad g^k=\frac{1}{n}\sum_{i=1}^n g_i^k$,

$$g_i^k=\begin{cases}\nabla f_i(x^k), & \text{with prob. } p,\\ g_i^{k-1}+\nabla f_{i,j_i^k}(x^k)-\nabla f_{i,j_i^k}(x^{k-1}), & \text{with prob. } 1-p\end{cases}$$

✓ Variances of the estimators g_i^k go to zero
✓ Analysis does not rely on the unbiasedness: $\mathbb{E}[g_i^k\mid x^k]\neq\nabla f_i(x^k)$

How can variance reduction help? *It leaves less space for Byzantines to hide in the noise.*



4. Ingredient 2: Clipping

Clipping operator:

$$\text{clip}(x,\lambda)=\begin{cases}\min\left\{1,\frac{\lambda}{\|x\|}\right\}x, & \text{if } x\neq 0,\\ 0, & \text{otherwise.}\end{cases}\tag{2}$$

Properties of clipping:

- Boundedness: $\|\text{clip}(x,\lambda)\|\le\lambda$
  ✓ If the direction is spoiled, clipping ensures that the algorithm does not go far away even when Byzantines form majority
- Controlled bias: $\|\text{clip}(x,\lambda)-x\|\le\left(1-\min\left\{1,\frac{\lambda}{\|x\|}\right\}\right)\|x\|$
  ✓ If the vector x is good enough, the right choice of the clipping level will not spoil the magnitude of the vector
  ✓ Clipping preserves the direction

5. New Method: Byz-VR-MARINA-PP

Algorithm Byz-VR-MARINA-PP

Input: starting point x^0, stepsize γ, minibatch size b, probability $p\in(0,1]$, number of iterations K, (δ,c)-$\texttt{ARAgg}$, clients' sample size $1\le C\le n$, clipping coefficients $\{\alpha_k\}_{k\ge1}$, direction g^0
for $k=0,1,\ldots,K-1$ **do**
  Get a sample from Bernoulli distribution: $c_k\sim\text{Be}(p)$
  Sample the set of clients $S_k\subseteq[n]$, $|S_k|=C$ if $c_k=0$; otherwise $S_k=[n]$
  Broadcast g^k, c_k to all workers
  for $i\in\mathcal{G}\cap S_k$ in parallel **do**
   $x^{k+1}=x^k-\gamma g^k$ and $\lambda_{k+1}=\alpha_{k+1}\|x^{k+1}-x^k\|$
   Set $g_i^{k+1}=\begin{cases}\nabla f_i(x^{k+1}), & \text{if } c_k=1,\\ g^k+\text{clip}_{\lambda_{k+1}}\left(\mathcal{Q}\left(\widehat{\Delta}_i(x^{k+1},x^k)\right)\right), & \text{otherwise,}\end{cases}$
   where $\widehat{\Delta}_i(x^{k+1},x^k)$ is a minibatched estimator of $\nabla f_i(x^{k+1})-\nabla f_i(x^k)$, $\mathcal{Q}(\cdot)$ for $i\in\mathcal{G}\cap S_k$ are computed independently
  end for
  if $c_k=1$ **then**
   $g^{k+1}=\texttt{ARAgg}\left(\{g_i^{k+1}\}_{i\in[n]}\right)$
  else
   $g^{k+1}=g^k+\texttt{ARAgg}\left(\left\{\text{clip}_{\lambda_{k+1}}\left(\mathcal{Q}\left(\widehat{\Delta}_i(x^{k+1},x^k)\right)\right)\right\}_{i\in S_k}\right)$
  end if
end for

- When $\alpha_k\equiv+\infty$ **Byz-VR-MARINA-PP** reduces to **Byz-VR-MARINA**
- $\mathcal{Q}$ is a compression operator
- Clipping level is proportional to $\|x^{k+1}-x^k\|$, which is the key to controlling the bias

6. Technical Preliminaries

Definition 2: Unbiased Compression

Operator $\mathcal{Q}:\mathbb{R}^d\to\mathbb{R}^d$ is called unbiased compressor/compression operator if there exists $\omega\ge0$ such that for any $x\in\mathbb{R}^d$

$$\mathbb{E}[\mathcal{Q}(x)]=x,\quad\mathbb{E}\left[\|\mathcal{Q}(x)-x\|^2\right]\le\omega\|x\|^2.\tag{3}$$

Assumptions

- Bounded aggregator: $\texttt{ARAgg}(x_1,\ldots,x_n)\le F\max_{i\in[n]}\|x_i\|$
- Smoothness and lower-boundedness: $\forall x,y\in\mathbb{R}^d$ we have $\|\nabla f(x)-\nabla f(y)\|\le L\|x-y\|$, $\|\nabla f_i(x)-\nabla f_i(y)\|\le L_i\|x-y\|$ for $i\in\mathcal{G}$ and $f_*=\inf_{x\in\mathbb{R}^d}f(x)>-\infty$
- ζ^2-heterogeneity: $\frac{1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)-\nabla f(x)\|^2\le\zeta^2\quad\forall x\in\mathbb{R}^d$
- Global Hessian variance assumption: $\frac{1}{G}\sum_{i\in\mathcal{G}}\|\nabla f_i(x)-\nabla f_i(y)\|^2-\|\nabla f(x)-\nabla f(y)\|^2\le L_\pm^2\|x-y\|^2$
- Local Hessian variance assumption: $\frac{1}{G}\sum_{i\in\mathcal{G}}\mathbb{E}\|\widehat{\Delta}_i(x,y)-\Delta_i(x,y)\|^2\le\frac{\mathcal{L}_\pm^2}{b}\|x-y\|^2$, where $\Delta_i(x,y)=\nabla f_i(x)-\nabla f_i(y)$ and $\widehat{\Delta}_i(x,y)$ is an unbiased mini-batched estimator of $\Delta_i(x,y)$ with batch size b

7. Convergence Results

Theorem 1

Let the introduced assumptions hold and $\lambda_{k+1}=2\max_{i\in\mathcal{G}}L_i\left\|x^{k+1}-x^k\right\|$. Assume that $0<\gamma\le\frac{1}{L+\sqrt{A}}$, where

$$\begin{aligned}A={}&\frac{4}{p}\left(\frac{80}{p}\frac{p_G\mathcal{P}_{\mathcal{G}_C^k}(1-\delta)n}{C^2(1-\delta_{\max})^2}\omega+\frac{4}{p}(1-p_G)+\frac{160}{p}p_G\mathcal{P}_{\mathcal{G}_C^k}c\delta_{\max}\omega\right)L^2+\frac{64}{p^2}(1-p_G)F_{\mathcal{A}}^2\max_{i\in\mathcal{G}}L_i^2\\&+\frac{4}{p}\left(\frac{8}{p}\frac{p_G\mathcal{P}_{\mathcal{G}_C^k}(1-\delta)n}{C^2(1-\delta_{\max})^2}(10\omega+1)+\frac{16}{p}p_G\mathcal{P}_{\mathcal{G}_C^k}c\delta_{\max}(10\omega+1)\right)L_\pm^2\\&+\frac{4}{p}\left(\frac{80}{p}\frac{p_G\mathcal{P}_{\mathcal{G}_C^k}(1-\delta)n}{C^2(1-\delta_{\max})^2}(\omega+1)+\frac{160}{p}p_G\mathcal{P}_{\mathcal{G}_C^k}c\delta_{\max}(\omega+1)\right)\frac{\mathcal{L}_\pm^2}{b},\end{aligned}$$

where $p_G:=\mathbb{P}\left\{G_C^k\ge(1-\delta_{\max})C\right\}$ and $\mathcal{P}_{\mathcal{G}_C^k}:=\mathbb{P}\left\{i\in\mathcal{G}_C^k\mid G_C^k\ge(1-\delta_{\max})C\right\}$. Then for all $K\ge0$ the point $\widehat{x}^K$ chosen uniformly at random from the iterates $x^0,x^1,\ldots,x^K$ produced by Byz-VR-MARINA-PP satisfies

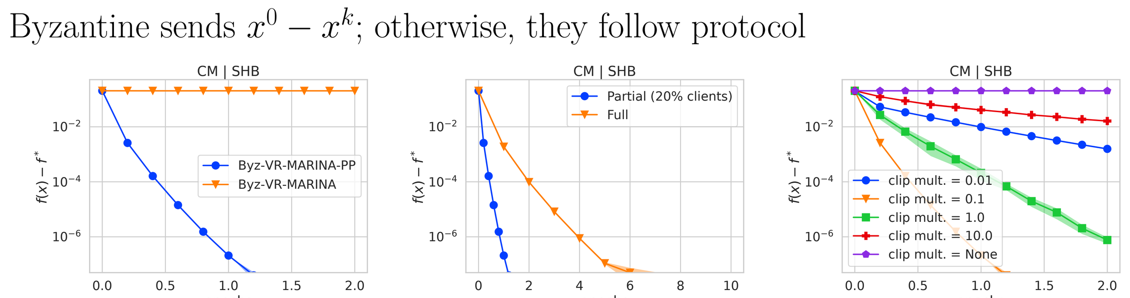
$$\mathbb{E}\left[\|\nabla f(\widehat{x}^K)\|^2\right]\le\frac{2\Phi_0}{\gamma(K+1)}+\frac{48c\delta\zeta^2}{p},\tag{4}$$

where $\Phi_0=f(x^0)-f_*+\frac{\gamma}{p}\|g^0-\nabla f(x^0)\|^2$ and $\mathbb{E}[\cdot]$ denotes the full expectation.

- When $\zeta=0$ (homogeneous data) the method converges asymptotically to the exact solution with rate $\mathcal{O}(^1/_K)$
- If $C=1$, then $p_G=\frac{G}{n}$ and $\mathcal{P}_{\mathcal{G}_C^k}=\frac{1}{G}$; if $C=2$, then $p_G=\frac{G(G-1)}{n(n-1)}$ and $\mathcal{P}_{\mathcal{G}_C^k}=\frac{2}{G}$; finally, if $C=n$, then $p_G=1$ and $\mathcal{P}_{\mathcal{G}_C^k}=1$
- Recommended value of $p=\min\{^C/_n,{}^b/_m,{}^1/_{(1+\omega)}\}$

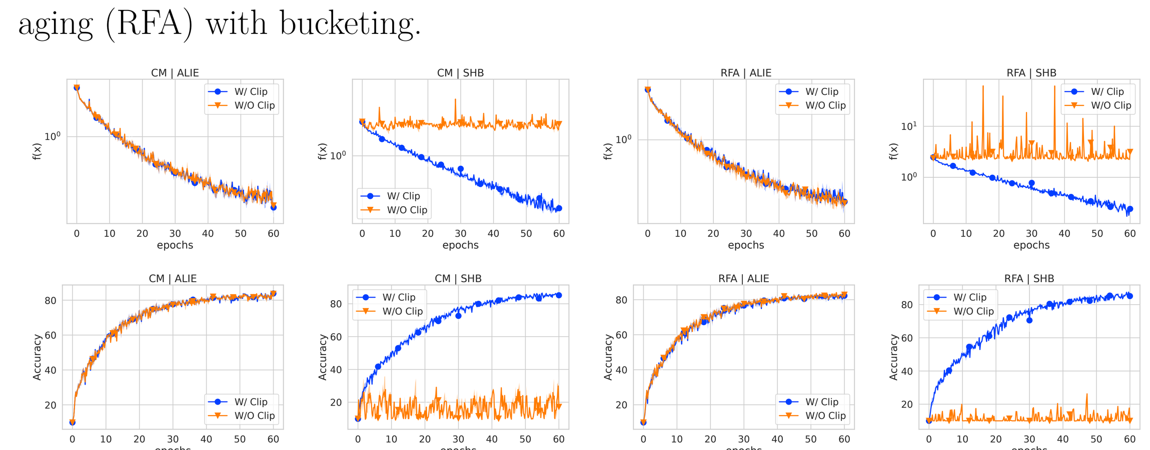
8. Experiments

- We consider a logistic regression model with ℓ_2-regularization
- 15 good workers and 5 Byzantines; *access to the entire dataset*
- Aggregation: coordinate-wise median with bucketing
- Shift-back attack: if Byzantines form a majority during round k, then each Byzantine sends x^0-x^k; otherwise, they follow protocol



Left: Linear convergence of Byz-VR-MARINA-PP with clipping versus non-convergence without clipping. Middle: Full versus partial participation. Right: Clipping multiplier α sensitivity.

- We consider a ResNet-18 model architecture with layer norm
- We consider the CIFAR 10 dataset with heterogeneous splits with 20 clients, 5 of which are Byzantines, and 4 clients are sampled in each step
- Attacks: we consider A Little is Enough (ALIE), Bit Flipping (BF), and Shift-Back (SHB) attacks
- Aggregation: we consider coordinate median (CM) and robust federated averaging (RFA) with bucketing.



Training loss (top) and test accuracy (bottom) of 2 aggregation rules (CM, RFA) under 4 attacks (BF, LF, ALIE, SHB) on the CIFAR10 dataset under heterogeneous data split with 20 clients, 5 of which are malicious, 4 clients sampled per round.

References

[1] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. ICLR 2022.
[2] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. NeurIPS 2014.
[3] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. ICML 2017.
[4] Samuel Horváth, Lihua Lei, Peter Richtárik, and Michael I. Jordan. Adaptivity of stochastic gradient methods for nonconvex optimization. SIAM Journal on Mathematics of Data Science, 2022.
[5] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. ICML 2021.
[6] Eduard Gorbunov, Konstantin P Burlachenko, Zhize Li, and Peter Richtárik. MARINA: Faster non-convex distributed learning with compression. ICML 2021.
[7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. NeurIPS 2017.
[8] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. NeurIPS 2019.
[9] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. UAI 2020.