Byzantine-Tolerant Methods for Distributed Variational Inequalities

Nazarii Tupitsa, Abdulla Jasem Almansoori, Yanlin Wu, Martin Takáč, Karthik Nandakumar, Samuel Horváth and Eduard Gorbunov

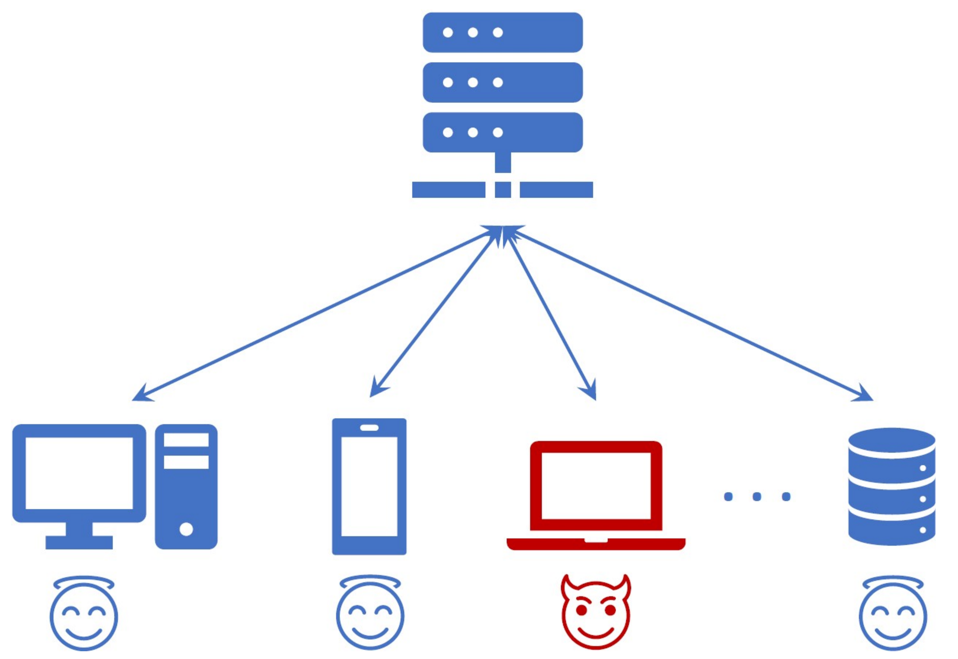
Mohamed bin Zayed University of Artificial Intelligence

Distributed VI problem

A lot of problems cannot be reduced to minimization, e.g., adversarial training [1], generative adversarial networks (GANs) [2], hierarchical reinforcement learning [3], adversarial examples games [4], problems arising in game theory, control theory, and differential equations [5]. Such problems lead to min-max or, more generally, variational inequality (VI) problems [6] that have significant differences from minimization ones (but include minimization) and require special consideration [7, 8].

$$\text{Find } \boldsymbol{x}^* \in \mathbb{R}^d \text{ s.t. } F(\boldsymbol{x}^*) = 0, \text{ where } F(\boldsymbol{x}) := \frac{1}{G}\sum_{i\in\mathcal{G}} F_i(\boldsymbol{x}),$$

- $\mathcal{G}$ is the set of good 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



Robust Aggregation

Popular aggregation rules:

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

These defenses are vulnerable to Byzantine attacks [9, 10] and do not satisfy the following definition.

Definition 1: (δ, c)-Robust Aggregator (modification of the definition from [11])

If a subset $\mathcal{G} \subseteq [n]$ of $\{x_1, x_2, \ldots, x_n\}$ is s.t. $|\mathcal{G}| = G \geq (1-\delta)n$ for $\delta < 0.5$ and there exists $\sigma \geq 0$ such that $\frac{1}{G(G-1)}\sum_{i,l\in\mathcal{G}} \mathbb{E}\|x_i - x_l\|^2 \leq \sigma^2$ where the expectation is taken w.r.t. the randomness of $\{x_i\}_{i\in\mathcal{G}}$, then $\widehat{x} = \texttt{RAgg}(x_1, \ldots, x_n)$ is called (δ, c)-**Robust Aggregator** ((δ, c)-$\texttt{RAgg}$) if the following holds:

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

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, we say that $\widehat{x}$ is (δ, c)-**Agnostic Robust Aggregator** ((δ, c)-$\texttt{ARAgg}$) and write $\widehat{x} = \texttt{ARAgg}(x_1, \ldots, x_n)$.

One can robustify $\texttt{Krum}$, $\texttt{RFA}$, and $\texttt{CM}$ using bucketing [11].

Algorithm 1 $\texttt{Bucketing}$: Robust Aggregation using bucketing [11]

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

Existing Methods

$\texttt{Parallel SGDA}$ / $\texttt{Parallel SEG}$:
- ✗ Permutation invariance
- ✗ Divergence with Byzantines

$\texttt{RDEG}$ [12]:
- ✗ Permutation invariance
- ✗ Convergence with *large batches* in **homogeneous** case

Why permutation non-invariance? *As [13] prove, any permutation-invariant algorithm fails to converge to any predefined accuracy even if workers have homogeneous data!*

Main Contribution

- **Methods with provably robust aggregation.** We propose new methods $\texttt{SGDA-RA}$ and $\texttt{SEG-RA}$ – variants of popular $\texttt{SGDA}$ and $\texttt{SEG}$. We prove that $\texttt{SGDA-RA}$ and $\texttt{SEG-RA}$ work with any (δ, c)-robust aggregation rule and converge to the desired accuracy *if the batchsize is large enough.*
- **Client momentum.** We add client momentum to $\texttt{SGDA-RA}$ and propose Momentum $\texttt{SGDA-RA}$ ($\texttt{M-SGDA-RA}$). That breaks the permutation invariance. In the case of star-cocoercive quasi-strongly monotone VIs, we prove the convergence to the neighborhood of the solution; the size of the neighborhood can be reduced via increasing batchsize only – similarly to the results for $\texttt{RDEG}$, $\texttt{SGDA-RA}$, and $\texttt{SEG-RA}$.
- **Methods with random checks of computations.** For homogeneous data case ($\zeta = 0$), we propose a version of $\texttt{SGDA}$ and $\texttt{SEG}$ with random checks of computations ($\texttt{SGDA-CC}$, $\texttt{SEG-CC}$ and their restarted versions – $\texttt{R-SGDA-CC}$ and $\texttt{R-SEG-CC}$). We prove that the proposed methods converge *to any accuracy of the solution without any assumptions on the batchsize.* Moreover, when the target accuracy of the solution is small enough, the obtained convergence rates for $\texttt{R-SGDA-CC}$ and $\texttt{R-SEG-CC}$ are not worse than the ones for distributed $\texttt{SGDA}$ and $\texttt{SEG}$ derived in the case of no Byzantine workers; see the comparison of the convergence rates in Table 1.

Methods with Robust Aggregation

$$\mathbb{E}_{\boldsymbol{\xi}_i}\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i) = F_i(\boldsymbol{x}) \qquad \mathbb{E}_{\boldsymbol{\xi}_i}\|\boldsymbol{g}_i(\boldsymbol{x}, \boldsymbol{\xi}_i) - F_i(\boldsymbol{x})\|^2 \leq \sigma^2. \tag{2}$$

$\texttt{SGDA-RA}$: $\quad \boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \gamma\text{RAgg}(\boldsymbol{g}_1^t, \ldots, \boldsymbol{g}_n^t),$

where $\boldsymbol{g}_i^t = \boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)\ \forall\, i \in \mathcal{G}$, $\boldsymbol{g}_i^t = *\ \forall\, i \in \mathcal{B}$, and $\{\boldsymbol{g}_i^t\}_{i\in\mathcal{G}}$ are sampled independently.
- ✗ Permutation non-invariance
- ✓ Convergence with large batches in **heterogeneous** case
- ✗ Convergence with large batches in **homogeneous** case

$\texttt{M-SGDA-RA}$: $\quad \boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \gamma\text{RAgg}(\boldsymbol{m}_1^t, \ldots, \boldsymbol{m}_n^t),$
with $\boldsymbol{m}_i^t = (1-\alpha)\boldsymbol{m}_i^{t-1} + \alpha\boldsymbol{g}_i^t,$

where $\boldsymbol{g}_i^t = \boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)$, $\forall i \in \mathcal{G}$ and $\boldsymbol{g}_i^t = *\ \forall\, i \in \mathcal{B}$ and $\{\boldsymbol{g}_{\boldsymbol{\xi}_i}^t\}_{i\in\mathcal{G}}$ are sampled independently.
- ✓ Permutation non-invariance
- ✓ Convergence with large batches in **heterogeneous** case
- ✗ Convergence with large batches in **homogeneous** case

Methods with Checks of Computations

Key idea of the checks

At each iteration of $\texttt{SGDA-CC}$, the server selects m workers (uniformly at random) and requests them to check the computations of other m workers from the previous iteration.

Let V_t be the set of workers that verify/check computations, A_t are active workers at iteration t, and $V_t \cap A_t = \varnothing$. Then, the update of $\texttt{SGDA-CC}$ can be written as

$\texttt{SGDA-CC}$:
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - \gamma\overline{\boldsymbol{g}}^t,$$
$$\text{if } \overline{\boldsymbol{g}}^t = \frac{1}{|A_t|}\sum_{i\in A_t} \boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t) \quad \text{is accepted,}$$

where $\{\boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)\}_{i\in\mathcal{G}}$ are sampled independently.
The acceptance (of the update) event occurs when the condition $\|\overline{\boldsymbol{g}}^t - \boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)\| \leq C\sigma$ holds for the majority of workers. If $\overline{\boldsymbol{g}}^t$ is rejected, then all workers re-sample $\boldsymbol{g}_i(\boldsymbol{x}^t, \boldsymbol{\xi}_i^t)$ until acceptance is achieved. The rejection probability is bounded, as per [14].
- ✗ Permutation invariance
- ✗ Non applicable for **heterogeneous** case
- ✓ Convergence with *any batches* in **homogeneous** case

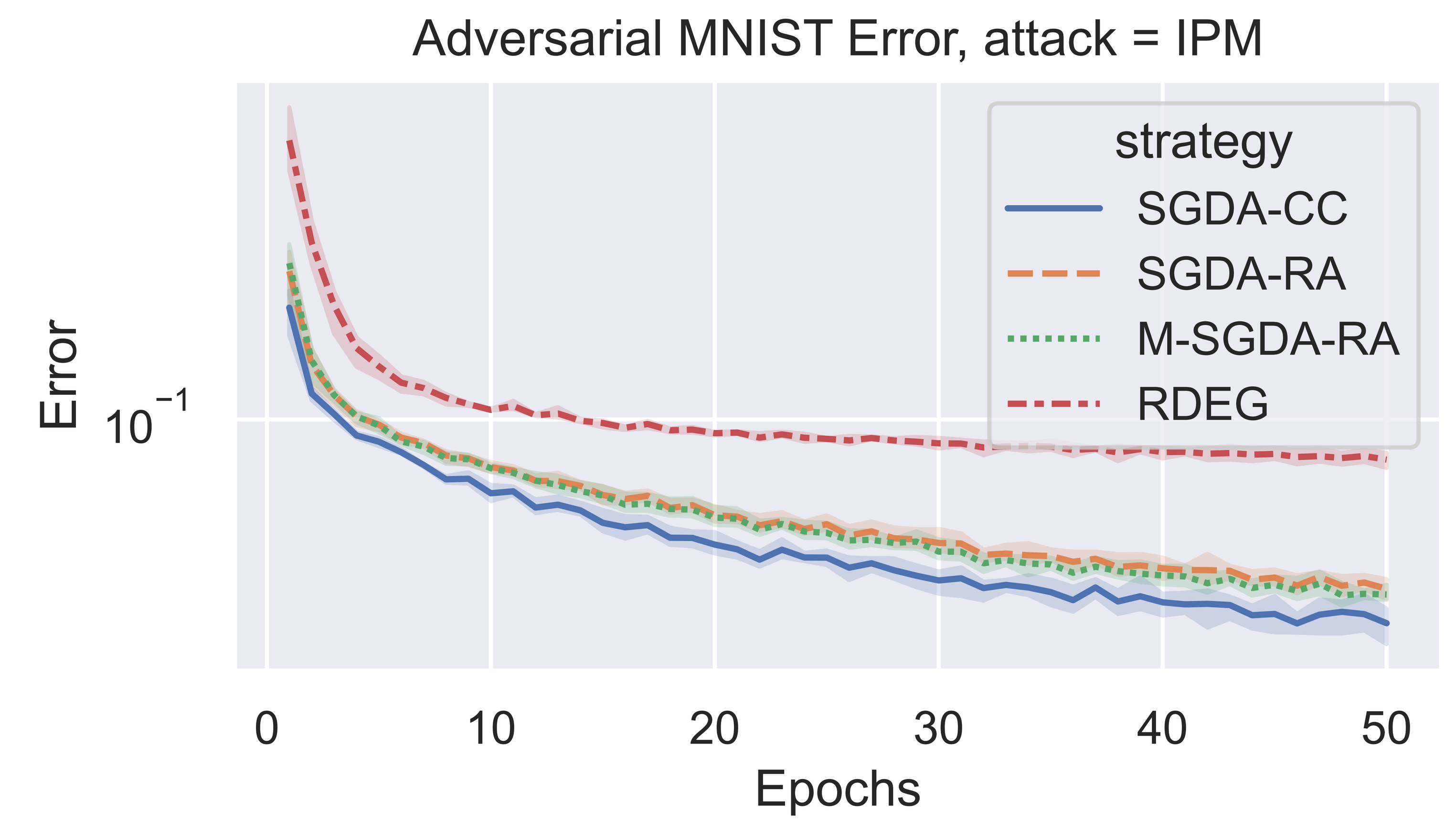
$\texttt{R-SGDA-CC}$: $\quad$ restarted version of $\texttt{SGDA-CC}$
- ✓ Additionally benefits of collaboration

Rates and Comparison

Table 1: By the complexity, we mean the number of stochastic oracle calls needed for a method to guarantee that $\mathbb{E}\|\boldsymbol{x}^T - \boldsymbol{x}^*\|^2 \leq \varepsilon$ (for $\texttt{RDEG}$ $\mathbf{P}\{\|\boldsymbol{x}^T - \boldsymbol{x}^*\|^2 \leq \varepsilon\} \geq 1 - \delta_{\texttt{RDEG}}$, $\delta_{\texttt{RDEG}} \in (0, 1]$). Column "BS" indicates the minimal batch-size used for achieving the corresponding complexity. Notation: c, δ are robust aggregator parameters; α = momentum parameter; β = ratio of inner and outer stepsize in $\texttt{SEG}$-like methods; n = total numbers of peers; m = number of checking peers; G = number of peers following the protocol; R = any upper bound on $\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|$; μ = quasi-strong monotonicity (QSM) parameter; ℓ = star-cocoercivity (SC) parameter; L = Lipschitzness (Lip) parameter; σ^2 = bound on the variance. The definition x^T can vary; see corresponding theorems for the exact formulas.

Setup	Method	Complexity	BS
(SC), (QSM)	$\texttt{SGDA-RA}$	$\frac{\ell}{\mu} + \frac{1}{c\delta n}$	$\frac{c\delta\sigma^2}{\mu^2\varepsilon}$
	$\texttt{M-SGDA-RA}$	$\frac{\ell}{\mu\alpha^2} + \frac{1}{c\delta\alpha n}$	$\frac{c\delta\sigma^2}{\alpha^2\mu^2\varepsilon}$
	$\texttt{SGDA-CC}$	$\frac{\ell}{\mu} + \frac{\sigma^2}{\mu^2 n\varepsilon} + \frac{\sigma^2 n^2}{\mu^2 m\varepsilon} + \frac{\sigma^2 n^2}{\mu^2 m\sqrt{\varepsilon}}$	1
	$\texttt{R-SGDA-CC}$	$\frac{\ell}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{n^2\sigma}{m\sqrt{\mu\varepsilon}}$	1
(Lip), (QSM)	$\texttt{SEG-RA}$	$\frac{L}{\beta\mu} + \frac{1}{\beta c\delta G} + \frac{1}{\beta}$	$\frac{c\delta\sigma^2}{\beta\mu^2\varepsilon}$
	$\texttt{SEG-CC}$	$\frac{L}{\mu} + \frac{1}{\beta} + \frac{\sigma^2}{\beta^2\mu^2 n\varepsilon} + \frac{\sigma^2 n^2}{\beta\mu^2 m\varepsilon} + \frac{\sigma^2 n^2}{\beta^2\mu^2 m\sqrt{\varepsilon}}$	1
	$\texttt{R-SEG-CC}$	$\frac{L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \frac{n^2\sigma}{m\sqrt{\mu\varepsilon}}$	1
(Lip), (QSM)	$\texttt{RDEG}$	$\frac{L}{\mu}$	$\frac{\sigma^2\mu^2 R^2}{L^4\varepsilon^2}$

- $\texttt{RDEG}$ is only for homogeneous case ($\zeta = 0$)



References

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *International Conference on Learning Representations* (2015).

[2] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems* 27 (2014).

[3] Greg Wayne and LF Abbott. "Hierarchical control using networks trained with higher-level forward models". In: *Neural computation* 26.10 (2014), pp. 2163–2193.

[4] Joey Bose et al. "Adversarial example games". In: *Advances in neural information processing systems* 33 (2020), pp. 8921–8934.

[5] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems.* Springer, 2003.

[6] Gauthier Gidel et al. "A Variational Inequality Perspective on Generative Adversarial Networks". In: *International Conference on Learning Representations.* 2018.

[7] Patrick T Harker and Jong-Shi Pang. "Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications". In: *Mathematical programming* 48.1-3 (1990), pp. 161–220.

[8] Ernest K Ryu and Wotao Yin. *Large-Scale Convex Optimization: Algorithms & Analyses via Monotone Operators.* Cambridge University Press, 2022.

[9] Moran Baruch, Gilad Baruch, and Yoav Goldberg. *A Little Is Enough: Circumventing Defenses For Distributed Learning.* 2019. arXiv: 1902.06156 [cs.LG].

[10] Cong Xie, Sanmi Koyejo, and Indranil Gupta. *Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation.* 2019. arXiv: 1903.03936 [cs.LG].

[11] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. "Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing". In: *International Conference on Learning Representations.* 2022.

[12] Arman Adibi et al. "Distributed statistical min-max learning in the presence of Byzantine agents". In: *2022 IEEE 61st Conference on Decision and Control (CDC).* IEEE. 2022, pp. 4179–4184.

[13] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. "Learning from history for byzantine robust optimization". In: *International Conference on Machine Learning.* PMLR. 2021, pp. 5311–5319.

[14] Eduard Gorbunov et al. "Secure distributed training at scale". In: *International Conference on Machine Learning.* PMLR. 2022, pp. 7679–7739.