Byzantine-Resilient Federated Alternating Gradient Descent and Minimization for Partly-Decoupled Low Rank Matrix Learning

International Conference on Machine Learning, 2025

Presenter Ankit Pratap Singh

Department of Electrical and Computer Engineering
Iowa State University

Specific ML class of problems: Partly-Decoupled Federated Low-Rank Matrix Learning

We developed Byzantine-resilient algorithm for a specific class of machine learning problems: **Partly-Decoupled Federated Low-Rank Matrix Learning.**

Problem Setting

Learn a low rank $r \ll n, q$ matrix $\mathbf{X}^* \in \Re^{n \times q}$ from measurements of the form

$$\mathbf{y}_k := \mathbf{A}_k \mathbf{x}_k^*, k \in [q].$$

The matrix $\mathbf{A}_k$ is defined differently for each problem:

- For **Low Rank Matrix Completion** (LRMC) problem it is a diagonal 1-0 matrix.
- For **Low Rank Columnwise Compressive Sensing** (LRCS) problem it is a random Gaussian matrix. This problem is also referred to as **multi-task representation learning** or **few-shot learning**.
- For **Low Rank Phase Retrieval** (LRPR) problem it is a random Gaussian matrix, but only the magnitudes of the measurements are observed i.e., $\mathbf{z}_k = |\mathbf{y}_k|$.
- **Initialization:** All these problems require initialization, which reduces to a subspace estimation or PCA problem.

Vertical Federation

$$\mathbf{y}_k = \mathbf{A}_k \mathbf{x}_k^*, k \in [q]$$

$$[\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_q^*] = \mathbf{X}^* = \mathbf{U}^* \mathbf{B}^* = \mathbf{U}^* [\mathbf{b}_1^*, \mathbf{b}_2^*, ..., \mathbf{b}_q^*]$$

Each node $\ell \in [L]$ observes a subset of columns $k \in \mathcal{S}_\ell \subset [q]$, $|\mathcal{S}_\ell| = \widetilde{q} < q$.

Solving this problem requires solving

$$\min_{\substack{\mathbf{U} \in \Re^{n \times r} \\ \mathbf{B} \in \Re^{r \times q}}} f(U, B) = \min_{\substack{\mathbf{U} \in \Re^{n \times r} \\ \mathbf{B} \in \Re^{r \times q}}} \sum_{k=1}^{q} \|\mathbf{y}_k - \mathbf{A}_k \mathbf{U} \mathbf{b}_k\|^2 \tag{1}$$

Theorem (Subspace-Median)

Subspace-Median Algorithm[1]

For a $\tau < 0.4$, *suppose that, for at least* $(1-\tau)L$ $\mathbf{U}_\ell$*'s*

$$\Pr\left(\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_\ell) \leq \delta\right) \geq 1 - p$$

then, with probability at least $1 - \exp(-L\psi(0.4-\tau, p))$,

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_{out}) \leq 23\delta.$$

[1] Singh & Vaswani, Byzantine-resilient federated pca and low rank column-wise sensing, IEEE TIT, 2024

Challenges in GD part

Challenges: Non Identical data

Since

$$\mathbb{E}[\nabla_\ell(\mathbf{U}_{t-1}, \mathbf{B}_\ell)] = m(\mathbf{X}_\ell - \mathbf{X}^*{}_\ell)\mathbf{B}_\ell{}^\top = m(\mathbf{U}\mathbf{B}_\ell - \mathbf{U}^*\mathbf{B}^*{}_\ell)\mathbf{B}_\ell{}^\top$$

Therefore,

$$\mathbb{E}[\nabla_\ell(\mathbf{U}_{t-1}, \mathbf{B}_\ell)] \neq \mathbb{E}[\nabla_{\ell'}(\mathbf{U}_{t-1}, \mathbf{B}_{\ell'})]$$

Bounded heterogeneity Assumption

$$\max_{\ell,\ell' \in [L]} \|\mathbf{B}^*_\ell - \mathbf{B}^*_{\ell'}\|_F^2 \leq G_B^2 \sigma_{\max}^{*2}$$

This assumption in turn implies that, for all $\ell, \ell' \in [L]$,

$$\|\mathbf{X}^*{}_\ell - \mathbf{X}^*{}_{\ell'}\|_F^2 = \|\mathbf{U}^*\mathbf{B}^*_\ell - \mathbf{U}^*\mathbf{B}^*_{\ell'}\|_F^2 \leq G_B^2 \sigma_{\max}^{*2}$$

All past work for heterogeneous setting assumes a bound on the difference between gradients from different good nodes, at each algorithm iteration [Assumption 2][2], [Assumption 1][3].

Using this assumption we can bound the additional term. And the convergence result depends on this heterogeneity factor.

---

[2]Data & Diggavi, Byzantine-resilient high-dimensional federated learning, IEEE TIT, 2023

[3]Allouah et al., Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity, AISTATS, 2023

Challenges: Incoherence of $\mathbf{U}^*$

Incoherence of $\mathbf{U}^*$: LRcCS/LRPR problem, does not require incoherence of $\mathbf{U}^*$. In LRMC, we need to ensure incoherence of $\mathbf{U}$ at every iteration. This is hard because $\mathbf{U}$ is updated using possibly non-incoherent gradients from GM or Krum. To handle this, we introduce a filtering step.

Algorithm 1 Byz-AltGDmin-LRMC

AltGDmin Initialization:
Nodes $\ell = 1, ..., L$
Calculate and Push $\mathbf{U}_{0\ell}$ to center
Central Server
Define set $\mathcal{I}_0 = \{\}$
for $\ell = 1$ to L **do**
  if $\|\mathbf{u}_{0\ell}^j\| \leq 1.5\mu\sqrt{\frac{r}{n}}$ for all $j \in [n]$ **then**
    Add ℓ to set $\mathcal{I}_0$
end for
$\mathbf{U}_0 \longleftarrow \text{Byz} - \text{SubspaceEstimation}\{\mathbf{U}_{0\ell}\}_{\ell \in \mathcal{I}_0}$
Push $\mathbf{U}_0$ to nodes.

Algorithm 2 Byz-AltGDmin-LRMC

AltGDmin Iterations:
for $t = 1$ to T **do**
  Nodes $\ell = 1, ..., L$
  Calculate and Push ∇_ℓ to center
  Central Server
  Define set $\mathcal{I}_t = \{\}$
  for $\ell = 1$ to L **do**
    Compute $\mathbf{U}_{temp} \leftarrow \mathbf{U}_{t-1} - \eta \nabla_\ell$
    if $\|\mathbf{u}^j_{temp}\| \leq (1 - \frac{0.4}{\tilde{\kappa}^2})\|\mathbf{u}^j_{t-1}\| + 1.4\mu\sqrt{\frac{r}{n}}$ for all $j \in [n]$ **then**
      Add ℓ to set $\mathcal{I}_t$
  end for
  $\nabla_{Kr/GM} = \mathrm{Krum/GM}\{\nabla_\ell\}_{\ell \in \mathcal{I}_t}$
  Compute $\mathbf{U}_t \leftarrow QR(\mathbf{U}_{t-1} - \eta \nabla_{Kr/GM})$
  Push $\mathbf{U}_t$ to nodes.
end for
Output $\mathbf{U}_T$.

Byzantine-resilient Vertically federated LRMC

Bounded heterogeneity: $\max_{\ell,\ell' \in [L]} \|\mathbf{B}_\ell^* - \mathbf{B}_{\ell'}^*\|_F^2 \leq G_B^2 \sigma_{\max}^{*2}$

Theorem

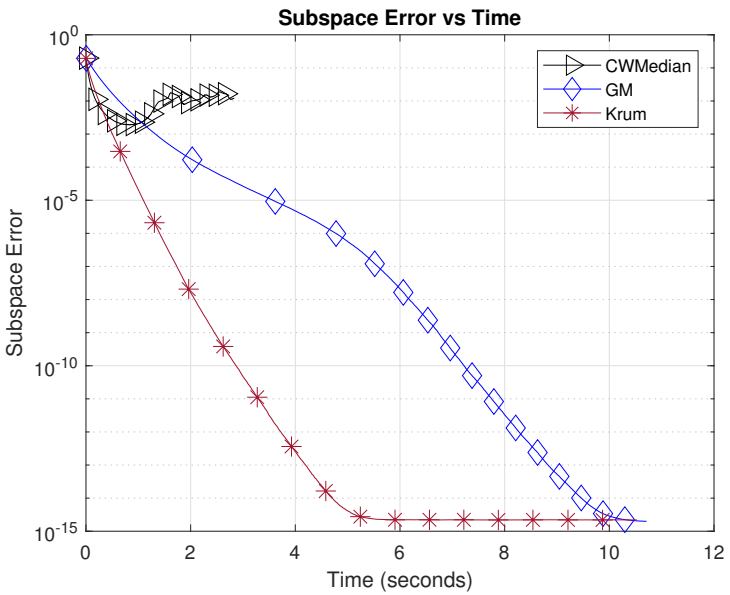
(Byz-Fed-AltGDmin-Learn: Complete guarantee) Assume RSV incoherence, Bounded heterogeneity Assumption holds, and $\frac{L_{byz}}{L} < 0.4$. *If*

$$n\widetilde{q}p \geq C\tilde{\kappa}^{10}\mu^2\widetilde{q}r^2 \log \widetilde{q} \log\left(\frac{1}{\epsilon}\right)$$

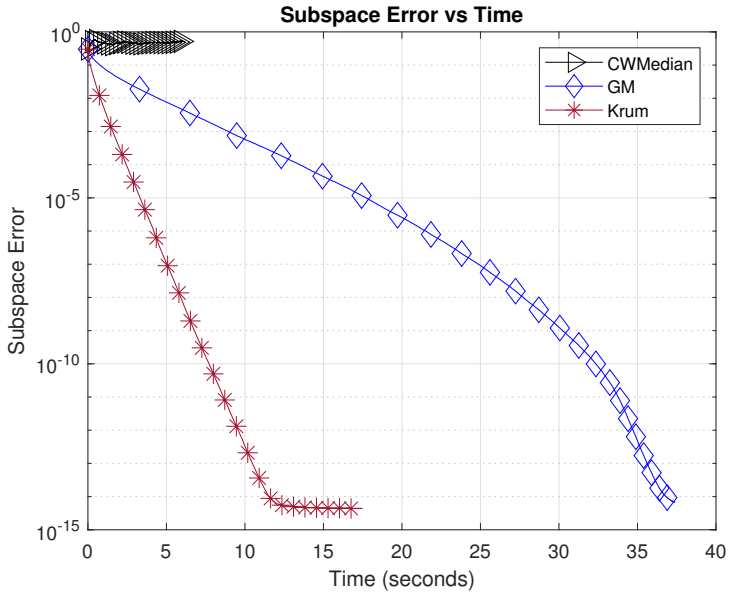
then, w.h.p. after $T = C\tilde{\kappa}^2 \log\left(\frac{1}{\epsilon}\right)$ *iterations,*

$$\mathbf{SD}_F(\mathbf{U}^*, \mathbf{U}_T) \leq \max(\epsilon, 14C\tilde{\kappa}^2 G_B)$$

Experiments



(a) **LRMC** with $n = 1000$, $q = 500$, $r = 3$, $L = 20$, and $L_{byz} = 8$.



(b) **LRCS** with $n = 1000$, $m = 50$, $q = 1000$, $r = 3$, $L = 20$, and $L_{byz} = 8$.

Figure 1: We compare Krum-AltGDmin, GM-AltGDmin, and CWMedian-AltGDmin for the different problems under the Reverse Gradient Attack.

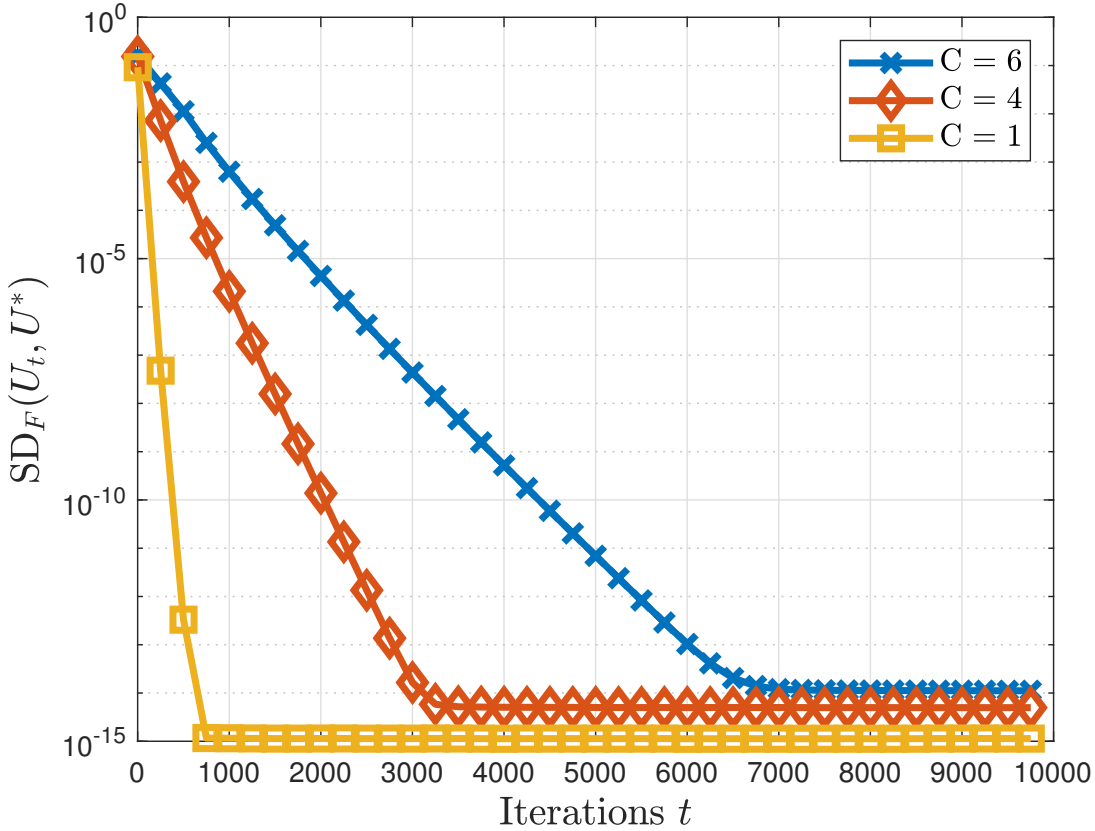


Figure 2: **Heterogeneity Effect: $\mathbf{SD}_F(\mathbf{U}_t, \mathbf{U}^*)$** vs Iteration t with $n = 200$, $q = 1000$, $r = 4, L = 10$, $L_{byz} = 2$, $p = 0.4$, Reverse Gradient Attack and using Krum

Comparison

Methods→	**Krum**	**GM**	**CWMed**
Sample Comp for Byz-AltGDmin (lower bound on $n\widetilde{q}p$)	$r^2\widetilde{q}\log\widetilde{q}\log(\frac{1}{\epsilon})$	$r^2\widetilde{q}\log\widetilde{q}\log(\frac{1}{\epsilon})$	$r\widetilde{q}\sqrt{n\log\widetilde{q}\log nr\log(\frac{1}{\epsilon})}$
Communic Cost	$nr\log(\frac{1}{\epsilon})$	$nr\log(\frac{1}{\epsilon})$	$nr\log(\frac{1}{\epsilon})$
Approximate Algorithm	No	Yes	No
Compute Cost at Center - GD	$nr^2L^2\log(\frac{1}{\epsilon})$	$nr^2L\log^3(\frac{L}{\epsilon_{approx}})\log(\frac{1}{\epsilon})$	$nr^2L\log(L)\log(\frac{1}{\epsilon})$
Compute Cost at Node - GD Ω is set of observed entries, $\mathbb{E}[\lvert\Omega\rvert] = npq$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$	$\max(n, \frac{\lvert\Omega\rvert}{L})r^2\log(\frac{1}{\epsilon})$

Table 1: *We compare Krum, Geometric Median (GM), and Coordinate wise median (CWMed) based modification of AltGDmin. Observe that Compute cost for CWMed is smallest but its sample complexity is unreasonably high making it useless. Krum and GM have same sample complexity. GM compute cost is slightly less than Krum but it is an approximate algorithm i.e., we can compute GM with* ϵ_{approx} *error.*

Thank You!