Robust Training in High Dimensions via Block Coordinate Geometric Median Descent

Anish Acharya, Abolfazl Hashemi,

Prateek Jain, Sujay Sanghavi, Inderjit Dhillon, Ufuk Topcu

THE UNIVERSITY OF TEXAS AT AUSTIN

PURDUE UNIVERSITY

Elmore Family School of Electrical and Computer Engineering

Google AI

AISTATS

Robust DNN Training

- Training DNN involves optimizing over highly over-parameterized , non-convex loss landscape.

- (Gross Corruption) Adversary can replace $0 \leq \psi \leq 1/2$ fraction of them with ***arbitrary*** points. If G and B are sets of good and bad points $\alpha = \frac{|B|}{|G|} = \frac{\psi}{\psi - 1} \leq 1$

- *smooth non-convex* problems with finite sum structure, under gross corruption, *without any prior knowledge about the malicious samples*.

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left[f(\mathbf{x}) := \frac{1}{|\mathbb{G}|} \sum_{i \in \mathbb{G}} f_i(\mathbf{x}) \right]$$

SGD under gross corruption

- SGD proceeds as follows : $\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \tilde{\mathbf{g}}^{(t)}, \quad \tilde{\mathbf{g}}^{(t)} = \frac{1}{|\mathcal{D}_t|} \sum_{i \in \mathcal{D}_t} \nabla f_i(\mathbf{x}_t).$

- Even a single corrupt sample can lead SGD to an *arbitrarily poor solution*.
  - This can be attributed to the *linear gradient aggregation* step.

- Breakdown Point: smallest fraction of contamination that must be introduced to cause an estimator to produce arbitrarily wrong estimates.

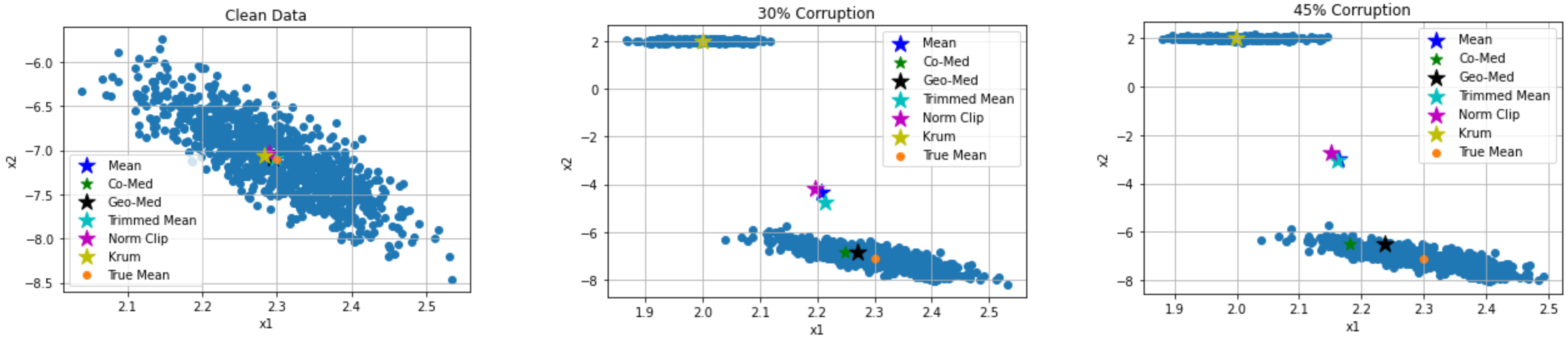
- SGD has lowest possible *asymptotic breakdown of 0* under gross corruption. Consider a single malicious gradient: $\mathbf{g}_j^{(t)} = -\sum_{i \in \mathcal{D}_t \setminus j} \mathbf{g}_i^{(t)}$

Robust Gradient Aggregation

- Make SGD Robust Again: Replace Mean with *Robust Mean Estimator*

- Geometric Median: $$\mathbf{x}_* = \textsc{Gm}(\{\mathbf{x}_i\}) = \arg\min_{\mathbf{y}\in\mathbb{X}} \left[g(\mathbf{x}) := \sum_{i=1}^{n} \|\mathbf{y} - \mathbf{x}_i\| \right]$$

- *Achieves Optimal Breakdown point of ½ .*



This Toy example in 2 dimensions demonstrates the superior robustness properties of GM for estimating the aggregated gradient even in presence of heavy corruption.

Geometric Median Descent

- GM-SGD : $x_{t+1} = \mathrm{x_t} - \eta\hat{g}_t$, $\hat{g}_t = \mathrm{GM}(\{\mathrm{g_i}\})$

- Unfortunately, finding GM is computationally hard.

- Best known algorithm to find $\epsilon -$ approximate GM i.e. $g(\mathbf{x}) \leq (1+\epsilon)g(\mathbf{x}_*)$ of n points in R^d requires $O(d/\epsilon^2)$.

- GM-SGD is computationally intractable for optimization in high dimensions arising from DNN e.g. $d \approx$ 60M Alexnet, $d \approx$ 175B GPT3

Block Coordinate GM Descent

- DNNs are over-parameterized
  - Performing *gradient aggregation in low dimensional subspace* should have *little impact* in the downstream optimization task.

- Judiciously *subset a block of k dimensions* (k << d) and perform GM in R^k
  - Ideally, select k dimensions resulting in largest decrease loss - NP Hard ☹
  - Select k columns with largest total norm from

- k << d can imply *large information loss* resulting in *slower convergence*.
  - Keep track of Residual and add back to gradient estimate.
  - Fixes sampling bias and retains convergence.

Theoretical Guarantee

- Non-convex and Smooth : Suppose f_i corresponding to non-corrupt samples i.e. $i \in G$ are L smooth and non-convex. Run BGMD with ϵ approximate GM oracle and $\gamma = \frac{1}{2L}$ in presence of α corruption for T iterations. Sample any iteration τ uniformly at random then:

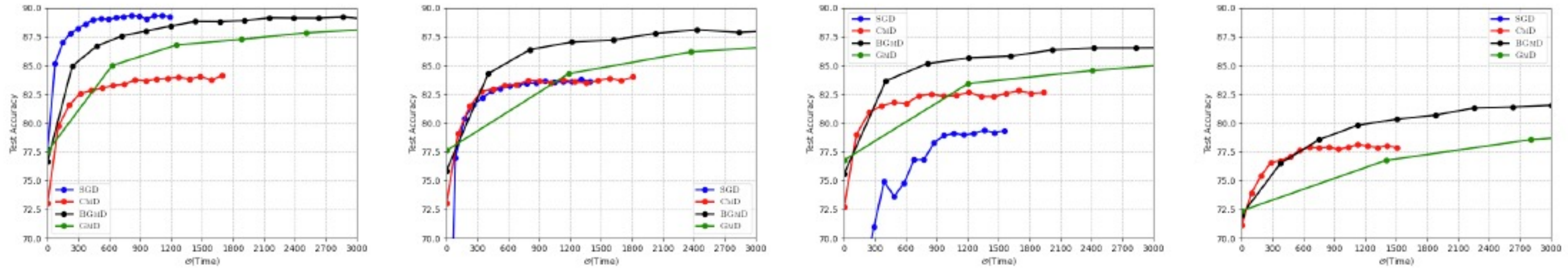
$$\mathbb{E}\|\nabla f(\mathbf{x}_\tau)\|^2 = \mathcal{O}\left(\frac{LR_0}{T} + \frac{\sigma^2\xi^{-2}}{(1-\alpha)^2} + \frac{L^2\epsilon^2}{|\mathbb{G}|^2(1-\alpha)^2}\right)$$

Algorithm	Aggregation Operator*	Iteration Complexity†	Breakdown Point†‡
SGD	MEAN(·)	$\boldsymbol{\mathcal{O}(bd)}$	0
(Yang et al., 2019; Yin et al., 2018)	CM(·)	$\mathcal{O}(bd\log b)$	**1/2**
(Wu et al., 2020)	GM(·)	$\mathcal{O}(d\epsilon^{-2}+bd)$	**1/2**
BGmD (This work)	BGM(·)	$\mathcal{O}(k\epsilon^{-2}+bd)$	**1/2**
(Data and Diggavi, 2020)	(Steinhardt et al., 2017)	$\mathcal{O}(db^2\min(d,b)+bd)$	1/4
(Blanchard et al., 2017)	KRUM(·)	$\mathcal{O}(b^2d)$	$\lfloor\beta\rfloor$
(Yin et al., 2018)	CM$_\beta$(·)	$\mathcal{O}(bd(1-2\beta)+bd\log b)$	$\lfloor\beta\rfloor$
(Ghosh et al., 2019; Gupta et al., 2020)	NC$_\beta$(·)	$\mathcal{O}(bd(2-\beta)+b\log b)$	$\lfloor\beta\rfloor$

Empirical Evidence : Feature Corruption

- Feature Corruption Simulation
  - *Huber's Contamination*: $z_t \sim \mathcal{N}(100, 1)$ directly added to the images.
  - *Impulse Corruption:* Salt and Pepper noise added by setting 90% of pixels to 0 or 1.
  - *Gaussian Blur:* Kernel size (5,5) and $\sigma = 100$.

Top (L: Clean, R: Huber's Contamination). Bottom(L: Impulse, R: Gaussian Blur).



(a) No corruption (b) 10% Corruption (c) 20% Corruption (d) 40% Corruption

Test accuracy as a ***function of wall clock time*** for training Fashion-MNIST using LeNet (1.16 M params) in presence of impulse noise.