Securing Distributed Machine Learning in High Dimensions

Jiaming Xu

The Fuqua School of Business
Duke University

Joint work with
Yudong Chen (Cornell) and Lili Su (MIT)

Workshop on Information, Learning and Decision
ShanghaiTech, July 1, 2018

Distributed Machine Learning: Robustness

- An attractive solution to large-scale problems
  - Algorithms: [Boyd et al. 11], [Jordan, Lee and Yang 16], etc.
  - Systems: [*Map-Reduce*, Dean and Ghemawat 08], etc.

Distributed Machine Learning: Robustness

- An attractive solution to large-scale problems
  - Algorithms: [Boyd et al. 11], [Jordan, Lee and Yang 16], etc.
  - Systems: [*Map-Reduce*, Dean and Ghemawat 08], etc.
- The necessity of robustness:

Distributed Machine Learning: Robustness

- An attractive solution to large-scale problems
  - Algorithms: [Boyd et al. 11], [Jordan, Lee and Yang 16], etc.
  - Systems: [*Map-Reduce*, Dean and Ghemawat 08], etc.
- The necessity of robustness: Corrupted data
  - Statistical noise: [Candes et al, JACM 11] [Loh and Wainwright, NIPS 11]
  - Adversarial corruption: No structural assumptions [Chen, Caramanis and Mannor, ICML 13] [Diakonikolas et al., FOCS 16] [Charikar et al., STOC 17]

Our Goal

- **Implicit assumption of previous work:** *Reliable* learning system
  - Each computing device follows some designed specification
- **Our focus:** *Unreliable* learning system
  - Adversarial attacks: Some *unknown* subset of computing devices are compromised, and behave adversarially – such as sending out malicious messages

Our Goal

- **Implicit assumption of previous work:** *Reliable* learning system
  - Each computing device follows some designed specification
- **Our focus:** *Unreliable* learning system
  - Adversarial attacks: Some *unknown* subset of computing devices are compromised, and behave adversarially – such as sending out malicious messages

Goal: Secure model training in *unreliable* learning system

Why consider unreliable learning system?

Privacy Risk in Conventional Learning Paradigm

- Data is collected from providers and stored at clouds

Privacy Risk in Conventional Learning Paradigm

- Data is collected from providers and stored at clouds
- Serious privacy risks:
  - Facebook data scandal
  - PRISM: Facebook, Google, Yahoo!, Apple, Microsoft, Dropbox, etc.

New Learning Paradigm: Federated Learning

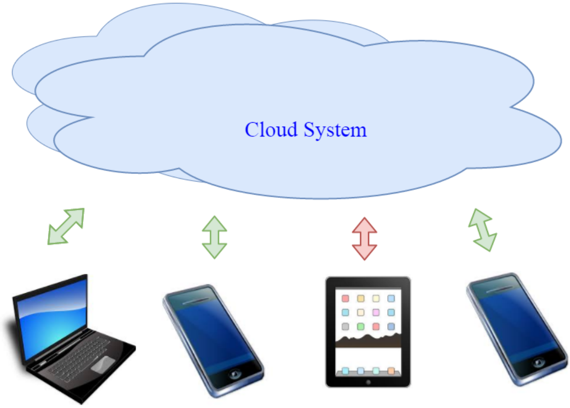
Key idea: Leave training data on mobile devices

- Learning with external workers (data providers)
- Proposed by Google researcher [McMahan 16]
- Tested by Gboard on Android and Google Keyboard

Leave training data on mobile devices

Security Risk in Federated Learning

- Less secured implementation environment
- External workers are prone to adversarial attack – reprogrammed by system hackers and behave maliciously



Leave training data on mobile devices

Goal: Secure model training in *unreliable* learning system

Challenges of Securing Unreliable Learning Systems

- Low local data volume versus high model complexity
  - Local estimator is statistically inaccurate
  - Hard to distinguish statistical errors from adversarial errors
  - Call for close interaction between the learner (cloud) and the workers
- Communication constraints: Data transmission suffers high latency and low throughout

Objectives

- Tolerate adversarial failures of the external workers
- Accurately learn highly complex models with low local data volume
- Use only a few communication rounds

Outline of the Remainder

1. Problem formulation
2. Algorithm 1: Geometric median of means
3. Algorithm 2 (Optimal Algorithm):
   Iterative rewriting + projecting + filtering
4. Summary and concluding remarks

Problem Formulation: Learning Model

- N i.i.d. data points $X_i \overset{i.i.d.}{\sim} \mu$
- Collectively kept by m workers – each worker keeps $\frac{N}{m}$ data points
- The learner wants to pick a model in $\Theta \subseteq \mathbb{R}^d$
- loss function $f(x, \theta)$: loss induced by $x \in \mathcal{X}$ under the model choice $\theta \in \Theta$

Target: $\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}[f(X, \theta)]$

NOTE: the population risk $F(\theta)$ is *unknown*

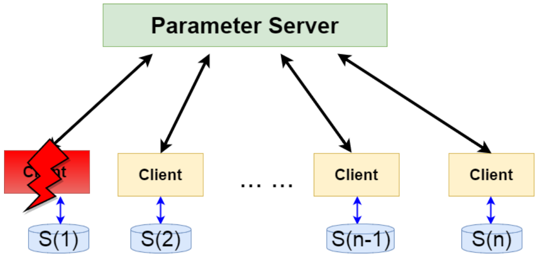
Example: Linear Regression

- N i.i.d. data points $X_i = (w_i, y_i) \overset{i.i.d.}{\sim} \mu$
  - w_i can be the features of a house/apartment, and y_i is its sold price
- $\Theta \subseteq \mathbb{R}^d$: the set of possible linear predictors
- Risk function $f(x, \theta) = \frac{1}{2}(y - \langle w, \theta \rangle)^2$

Target: $\theta^* \in \arg\min_{\theta \in \Theta} \mathbb{E}\left[\frac{1}{2}(y - \langle w, \theta \rangle)^2\right]$

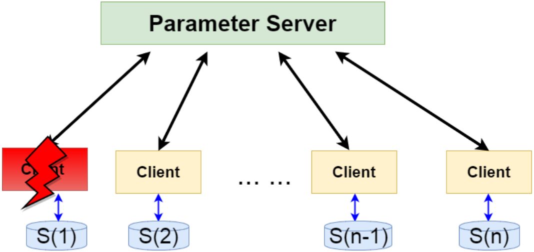
Problem Formulation: Byzantine Fault Model

- In any iteration, up to q out of m workers are compromised and behave arbitrarily;



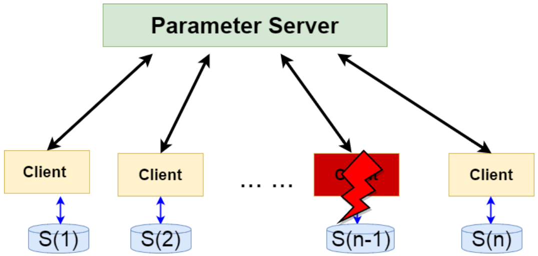
Problem Formulation: Byzantine Fault Model

- In any iteration, up to q out of m workers are compromised and behave arbitrarily;
- the set of faulty workers may be different across iterations;



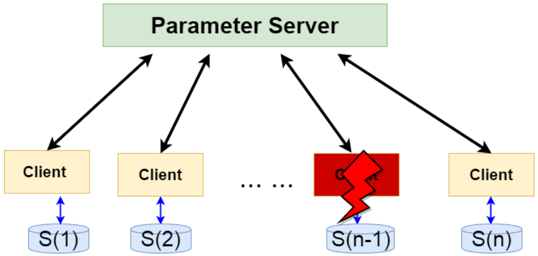
Problem Formulation: Byzantine Fault Model

- In any iteration, up to q out of m workers are compromised and behave arbitrarily;
- the set of faulty workers may be different across iterations;



Problem Formulation: Byzantine Fault Model

- In any iteration, up to q out of m workers are compromised and behave arbitrarily;
- the set of faulty workers may be different across iterations;
- faulty workers have complete knowledge of the system;
- faulty workers can collude



Algorithm: Byzantine Gradient Descent

The learner:

1. Broadcast the current model parameter estimator θ_{t-1};
2. Wait to receive all the gradients $g_t^{(j)}$ from all workers j;
3. Aggregate gradients to obtain $\hat{F}(\theta_{t-1})$;
4. Update: $\theta_t \leftarrow \theta_{t-1} - \eta_t \times \hat{F}(\theta_{t-1})$;

Non-faulty worker j:

1. Compute the sample gradient $g_t^{(j)} = \sum_{\text{local data } X_i} \nabla f(X_i, \theta_{t-1})$;
2. Send $g_t^{(j)}$ back to the learner;

Algorithm: Byzantine Gradient Descent

The learner:

1. Broadcast the current model parameter estimator θ_{t-1};
2. Wait to receive all the gradients $g_t^{(j)}$ from all workers j;
3. Aggregate gradients to obtain $\hat{F}(\theta_{t-1})$;
4. Update: $\theta_t \leftarrow \theta_{t-1} - \eta_t \times \hat{F}(\theta_{t-1})$;

Non-faulty worker j:

1. Compute the sample gradient $g_t^{(j)} = \sum_{\text{local data } X_i} \nabla f(X_i, \theta_{t-1})$;
2. Send $g_t^{(j)}$ back to the learner;

Averaging, i.e., taking $\hat{F}(\theta_{t-1}) = \frac{1}{m} \sum_{j=1}^{m} g_t^{(j)}$, is not robust to even a single Byzantine failure!

Algorithm: Byzantine Gradient Descent

The learner:

1. Broadcast the current model parameter estimator θ_{t-1};
2. Wait to receive all the gradients $g_t^{(j)}$ from all workers j;
3. **Robust gradient aggregate** to obtain $\hat{F}(\theta_{t-1})$;
4. Update: $\theta_t \leftarrow \theta_{t-1} - \eta_t \times \hat{F}(\theta_{t-1})$;

Non-faulty worker j:

1. Compute the sample gradient $g_t^{(j)} = \sum_{\text{local data } X_i} \nabla f(X_i, \theta_{t-1})$;
2. Send $g_t^{(j)}$ back to the learner;

Simple averaging, i.e., taking $\hat{F}(\theta_{t-1}) = \frac{1}{m} \sum_{j=1}^{m} g_t^{(j)}$, is not robust to even a single Byzantine failure!

Generic Key Technical Challenges

Target: $\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}[f(X,\theta)]$

- Suppose $F(\theta)$ is known: Perfect gradient descent –
  $\theta_t = \theta_{t-1} - \eta \times \nabla F(\theta_{t-1})$
- But $F(\theta)$ is unknown: Approximate gradient descent –

$$\theta'_t = \theta'_{t-1} - \eta_t \times \nabla \hat{F}(\theta'_{t-1}) = \theta'_{t-1} - \eta_t \times \nabla F(\theta'_{t-1}) + \epsilon(\theta'_{t-1}).$$

  - The elements in $\left\{\epsilon(\theta'_{t-1})\right\}_{t=1}^{\infty}$ are dependent on each other;
  - Complicated interplay between the randomness and the arbitrary behaviors of Byzantine workers.

Generic Key Technical Challenges

Target: $\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}[f(X, \theta)]$

- Suppose $F(\theta)$ is known: Perfect gradient descent –
  $\theta_t = \theta_{t-1} - \eta \times \nabla F(\theta_{t-1})$
- But $F(\theta)$ is unknown: Approximate gradient descent –

$$\theta'_t = \theta'_{t-1} - \eta_t \times \nabla \hat{F}(\theta'_{t-1}) = \theta'_{t-1} - \eta_t \times \nabla F(\theta'_{t-1}) + \epsilon(\theta'_{t-1}).$$

  - The elements in $\left\{\epsilon(\theta'_{t-1})\right\}_{t=1}^{\infty}$ are dependent on each other;
  - Complicated interplay between the randomness and the arbitrary behaviors of Byzantine workers.

Our analysis plan: show uniform convergence,

i.e., show $\epsilon(\theta) \approx 0$ uniformly for all $\theta \in \Theta$

Standard concentration results might not suffice

Algorithm I: Median of Means

Robust Gradient Aggregation: Median of Means

Median of Means

Given nk points $X_1, \ldots, X_{nk}$,

$$\hat{\phi}_{MM} \triangleq \text{median}\left\{\frac{1}{n}\sum_{i=1}^{n} X_i, \;\cdots,\; \frac{1}{n}\sum_{i=(k-1)n+1}^{kn} X_i\right\}$$

Definition (Geometric median)

$y^* \triangleq \text{med}\,\{y_1, \cdots, y_m\} = \arg\min_{y\in\mathbb{R}^d} \sum_{i=1}^{m} \|y - y_i\|_2$

Efficient computation of Geometric Median: Nearly linear time

[Cohen et al. STOC 2016]

Robustness of Geometric Median

Definition (Geometric median)

$y^* \triangleq \text{med}\{y_1, \cdots, y_m\} = \arg\min_{y \in \mathbb{R}^d} \sum_{i=1}^{m} \|y - y_i\|_2$

- **One-dimension case**: Geometric median $=$ standard median
  If strictly more than $\lfloor n/2 \rfloor$ points are in $[-r,\ r]$ for some $r \in \mathbb{R}$, then median **ALSO** lies in $[-r,\ r]$

Robustness of Geometric Median

Definition (Geometric median)

$y^* \triangleq \text{med}\{y_1, \cdots, y_m\} = \arg\min_{y \in \mathbb{R}^d} \sum_{i=1}^{m} \|y - y_i\|_2$

- **One-dimension case**: Geometric median = standard median
  If strictly more than $\lfloor n/2 \rfloor$ points are in $[-r, \ r]$ for some $r \in \mathbb{R}$, then median **ALSO** lies in $[-r, \ r]$
- **Multi-dimension case**:

Lemma (Minsker et al. 2015)

For any $\alpha \in (0, 1/2)$ *and given* $r \in \mathbb{R}$, *if* $\sum_{i=1}^{n} \mathbf{1}_{\{\|y_i\|_2 \le r\}} \ge (1-\alpha)n$, *then* $\|y_*\|_2 \le C_\alpha r$, *where* $C_\alpha = \frac{1-\alpha}{\sqrt{1-2\alpha}}$.

Robustness of Geometric Median

Definition (Geometric median)

$y^* \triangleq \text{med}\{y_1, \cdots, y_m\} = \arg\min_{y \in \mathbb{R}^d} \sum_{i=1}^{m} \|y - y_i\|_2$

- **One-dimension case**: Geometric median = standard median
  If strictly more than $\lfloor n/2 \rfloor$ points are in $[-r, \ r]$ for some $r \in \mathbb{R}$, then median **ALSO** lies in $[-r, \ r]$
- **Multi-dimension case**:

Lemma (Minsker et al. 2015)

For any $\alpha \in (0, 1/2)$ *and given* $r \in \mathbb{R}$, *if* $\sum_{i=1}^{n} \mathbf{1}_{\{\|y_i\|_2 \le r\}} \ge (1-\alpha)n$, *then* $\|y_*\|_2 \le C_\alpha r$, *where* $C_\alpha = \frac{1-\alpha}{\sqrt{1-2\alpha}}$.

Intuition: Majority voting in the noisy setting

Performance with Median of Means

(1) $q \geq 1$: the maximum # of Byzantine workers;
(2) d: model dimension, i.e., $\Theta \subseteq \mathbb{R}^d$

Theorem (Informal)

Suppose some mild technical assumptions hold, and $2(1+\epsilon)q \leq k \leq m$. Assume $F(\theta)$ is M-strongly convex with L-Lipschitz gradient. Then whp

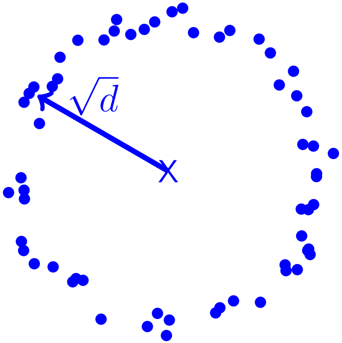
$$\|\theta_t - \theta^*\| \leq \rho^t \|\theta_0 - \theta^*\| + C\sqrt{\frac{dk}{N}}, \quad \forall t \geq 1,$$

where $\rho = \frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{M^2}{4L^2}} \in (0,1)$.

- After $\log N$ rounds, $\sqrt{dq/N}$ becomes the dominant part
- When $q = 0$, we choose $k = 1$
- When q is large, we choose $k = 2(1+\epsilon)q$, resulting error of $O(\sqrt{dq/N})$

Drawbacks of Geometric Median in High Dimensions

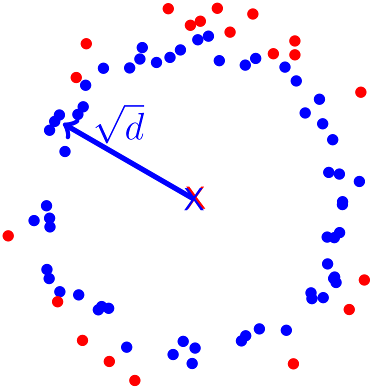
$$y^* = \arg\min \sum_{i=1}^{m} \|y - y_i\| \iff \sum_{i=1}^{m} \frac{y_i - y^*}{\|y_i - y^*\|} = \mathbf{0}$$



- Good data $y_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \mathbf{I}_d)$
- ϵ fraction is adversarially corrupted
- GM suffers from $\epsilon\sqrt{d}$ error

Drawbacks of Geometric Median in High Dimensions

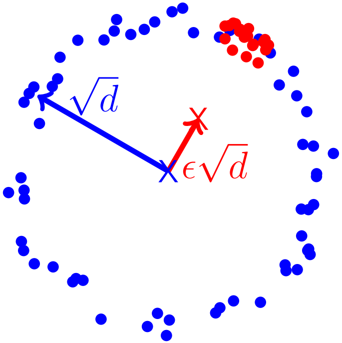
$$y^* = \arg\min \sum_{i=1}^{m} \|y - y_i\| \iff \sum_{i=1}^{m} \frac{y_i - y^*}{\|y_i - y^*\|} = \mathbf{0}$$



- Good data $y_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \mathbf{I}_d)$
- ϵ fraction is adversarially corrupted
- GM suffers from $\epsilon\sqrt{d}$ error

Drawbacks of Geometric Median in High Dimensions

$$y^* = \arg\min \sum_{i=1}^{m} \|y - y_i\| \iff \sum_{i=1}^{m} \frac{y_i - y^*}{\|y_i - y^*\|} = \mathbf{0}$$

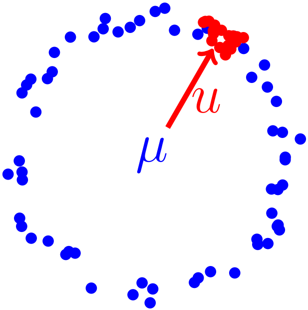


- Good data $y_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \mathbf{I}_d)$
- ϵ fraction is adversarially corrupted
- GM suffers from $\epsilon\sqrt{d}$ error

Algorithm II:
Optimal Algorithm in High Dimension

[Su and Xu, 2018] improves the estimation error from $O\left(\sqrt{\frac{qd}{N}}\right)$ to $O(\sqrt{d/N} + \sqrt{q/N})$ – matching the minimax error rate in the ideal failure-free setting as long as $q = O(d)$.

Ideas of Iterative Filtering [SCV '18]



- If the center μ were known, from

$$uu^\top \in \arg\max \sum_i (y_i - \mu)^\top U (y_i - \mu)$$
$$\text{s.t. } U \succeq 0$$
$$\mathrm{Tr}(U) \le 1,$$

  filter out outliers based on $\langle y_i - \mu, u\rangle^2$

- However, μ is unknown!
- Idea: represent y_i through $\sum_j W_{ji} y_j$; W_{ji} is constrained to be around $\frac{1}{(1-\epsilon)m}$

Iterative Filtering Algorithm [SCV '18]

Define cost function

$$\phi(W,U) = \sum_{i\in\mathcal{S}} c_i \left(y_i - \sum_{j\in\mathcal{S}} W_{ji} y_j \right)^\top U \left(y_i - \sum_{j\in\mathcal{A}} W_{ji} y_j \right)$$

1. Compute saddle point

$$\text{(Center approxi.)} \qquad W^* \in \arg\min_W \max_U \ \phi(W,U)$$

$$\text{(Extreme direction)} \qquad U^* \in \arg\max_U \min_W \ \phi(W,U)$$

2. If $\phi(W^*,U^*)$ is small enough, stop; otherwise, down-weight c_i proportional to $\left(y_i - \sum_{j\in\mathcal{S}} W^*_{ji} y_j\right)^\top U^* \left(y_i - \sum_{j\in\mathcal{S}} W^*_{ji} y_j\right)$, throw away data points for which $c_i \le 1/2$, and repeat.

Guarantees of Iterative Filtering Algorithm [SCV '18]

Lemma (SCV '18)

Define $\mu_{\mathcal{S}} = \frac{1}{m} \sum_{i=1}^{m} y_i$. *Suppose that*

$$\left\| \frac{1}{m} \sum_{i} \left(y_i - \mu_{\mathcal{S}}\right) \left(y_i - \mu_{\mathcal{S}}\right)^{\top} \right\|_2 \le \sigma^2.$$

Then for $\epsilon \le \frac{1}{4}$, *Iterative Filtering Algorithm outputs* $\hat{\mu}$ *such that*

$$\|\hat{\mu} - \mu_{\mathcal{S}}\| = O(\sigma\sqrt{\epsilon}).$$

- Gradient vectors $\{g_j(\theta_{t-1})\}_{j=1}^{m}$ are not i.i.d.
- Apply with $y_j =$ gradient functions:

$$g_j(\theta) = \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \nabla f\left(X_i,\ \theta\right)$$

- Need concentration of matrix $[g_1(\theta), \ldots, g_m(\theta)]$ uniformly over θ

Uniform Concentration of Sample Covariance Matrix

- If gradient functions $g_j(\theta)$ is sub-Gaussian, use ϵ-net
- However, in many cases such as linear regression, $g_j(\theta)$ is sub-exponential
- Existing tail bounds for matrices with sub-exponential columns are *not tight*
  State-of-the-art: Standard concentration bounds [ALPTJ '10]: $\sqrt{md} + d$

Theorem (SX '18)

Let A be a $d \times m$ matrix whose columns A_j are i.i.d. sub-exponential, zero-mean. Then with probability at least $1 - e^{-d}$,

$$\|A\|_2 \lesssim \sqrt{m} + d \log^3 d$$

Remark: Tight up to poly-log factors

Guarantees of Aggregated Gradient by Iterative Filtering

Theorem (SX '18)

Suppose some mild technical assumptions hold and $N \gtrsim d^2$*. Let* $\nabla \hat{F}(\theta)$ *be the aggregated gradient function by Iterative Filtering Algorithm. Then with probability at least* $1 - 2e^{-\sqrt{d}}$*,*

$$\left\| \nabla \hat{F}(\theta) - \nabla F(\theta) \right\| \lesssim \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \right) \|\theta - \theta^*\| + \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}} \right)$$

- $N \gtrsim d^2$ is due to our sub-exponential assumption and is inevitable
- If assuming sub-Gaussian instead, only $N \gtrsim d$ is needed

Main Convergence Result

Theorem (SX '18)

Suppose some mild technical assumptions hold and $N \gtrsim d^2$. Assume $F(\theta)$ is M-strongly convex with L-Lipschitz gradient. Then whp,

$$\|\theta_t - \theta^*\| \lesssim \left(1 - \frac{M^2}{16L^2}\right)^t \|\theta_0 - \theta^*\| + \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right).$$

- Improves over geometric median ($\sqrt{dq/N}$)
- If $q = O(d)$, error rate is optimal
- Tolerate up to $q/m = \Theta(1)$ fraction of Byzantine errors
- Exponential convergence $\rightarrow$ only logarithmic communication rounds

References

- Lili Su and Jiaming Xu,
  Securing Distributed Machine Learning in High Dimensions,
  arXiv:1804.10140, April 2018

- Yudong Chen, Lili Su, Jiaming Xu:
  Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent

  - Conference version: SIGMETRICS 2018;
  - Journal version: POMACS Proceedings of the ACM on Measurement and Analysis of Computing Systems, Dec. 2017.