Resilient distributed machine learning: Secure multi-agent federation

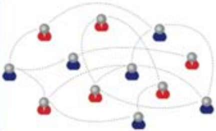
Lili Su (Northeastern, ECE)

CCAC21

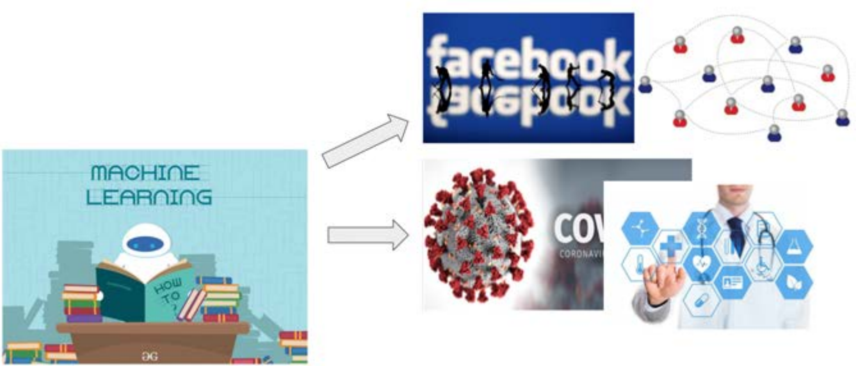
2021

Machine Learning

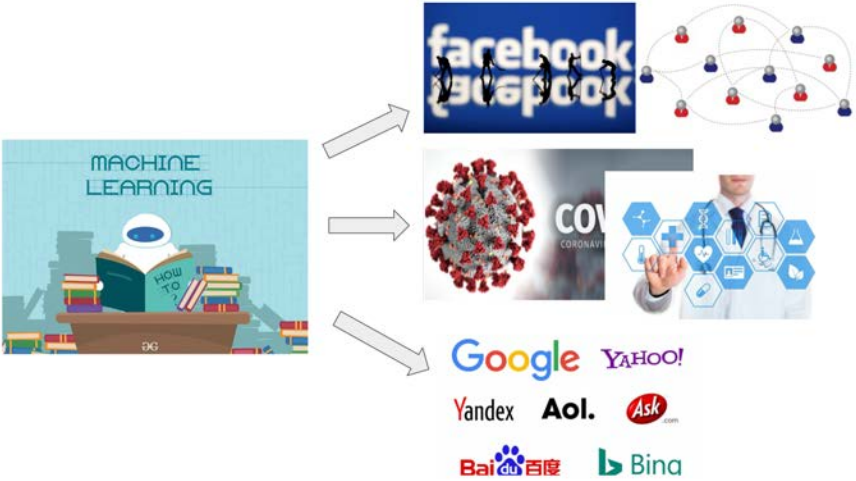
Machine Learning



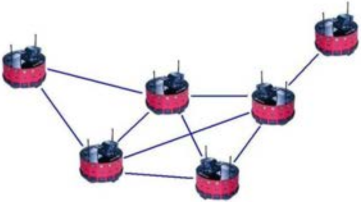
Machine Learning



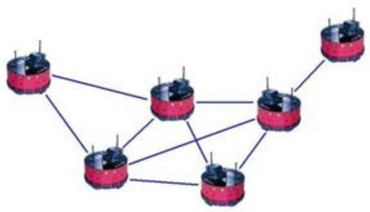
Machine Learning



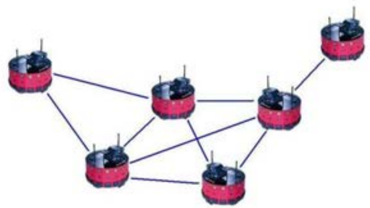
Large-scale Machine Learning

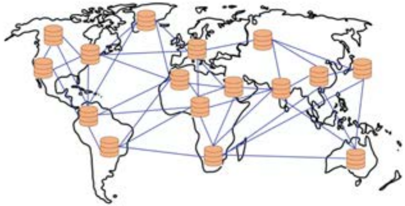


Large-scale Machine Learning

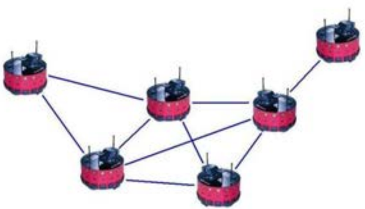


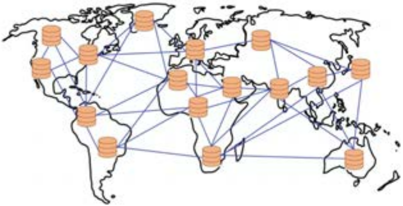
Large-scale Machine Learning





Large-scale Machine Learning





Large-scale Machine Learning

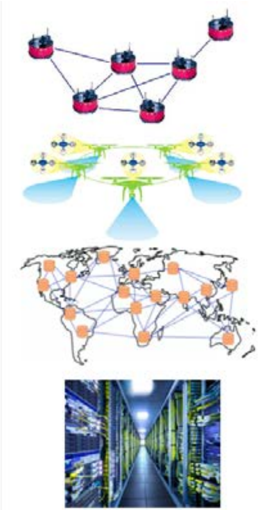
Cooperatively carry out a global task

Challenges

Challenges

Data heterogeneity: data collected at different devices might generate from different distributions

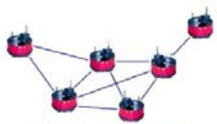
Challenges



Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Challenges



Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Privacy: data moving constraints

Challenges

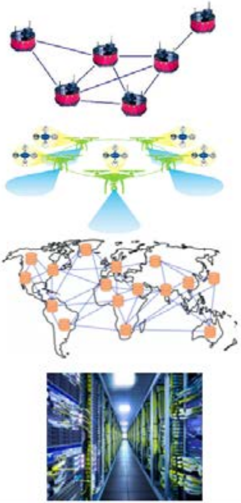
Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Privacy: data moving constraints

Lack of centralized data fusion center: data volume is so high that not a single machine is capability of heading of data fusion task

Challenges



Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Privacy: data moving constraints

Lack of centralized data fusion center: data volume is so high that not a single machine is capability of heading of data fusion task

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance

Challenges

Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Privacy: data moving constraints

Lack of centralized data fusion center: data volume is so high that not a single machine is capability of heading of data fusion task

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance

Popular System Architectures

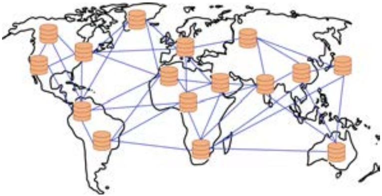
- Master-slave

Master: the cloud; Slaves: mobile devices

Popular System Architectures

- Master-slave
- Fully distributed

Master: the cloud; Slaves: mobile devices

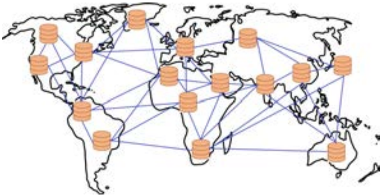


Fully distributed

Popular System Architectures

- Master-slave
- Fully distributed
- Hierarchical

Master: the cloud; Slaves: mobile devices



Fully distributed

Our focus

Master-slave

Fully distributed

Hierarchical

Data heterogeneity: data collected at different devices might generate from different distributions

Low local data volume: a device has limited data collection capability

Privacy: data moving constraints

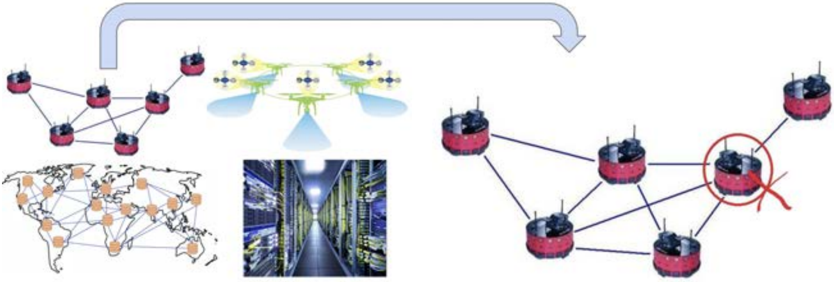
Lack of centralized data fusion center: data volume is so high that not a single machine is capability of heading of data fusion task

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance

Why adversary-resilient?

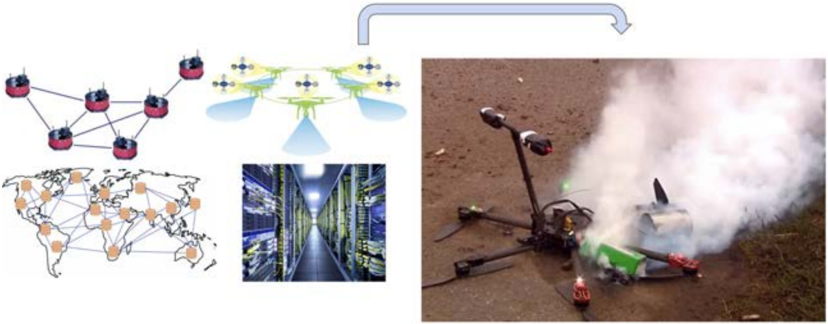
On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



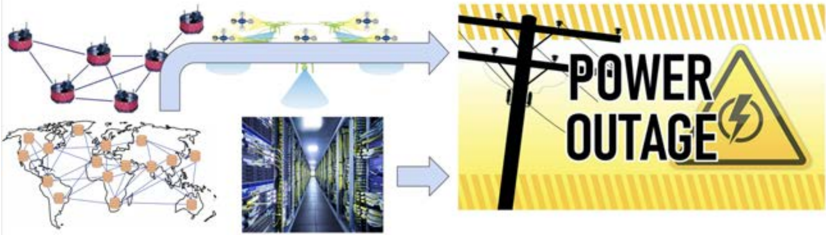
On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



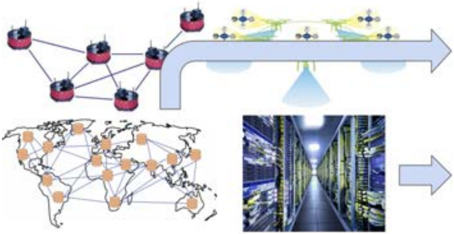
On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



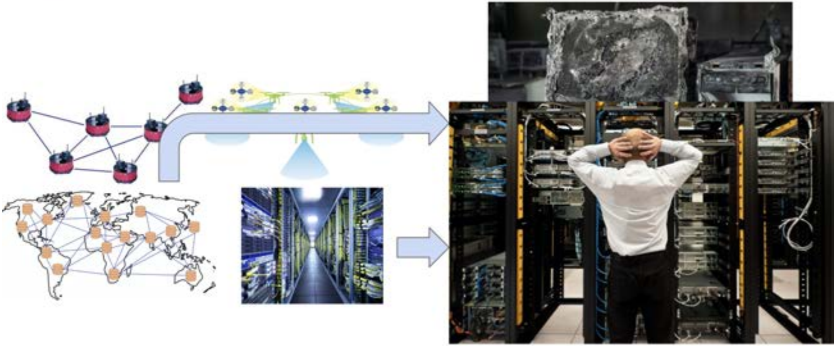
On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



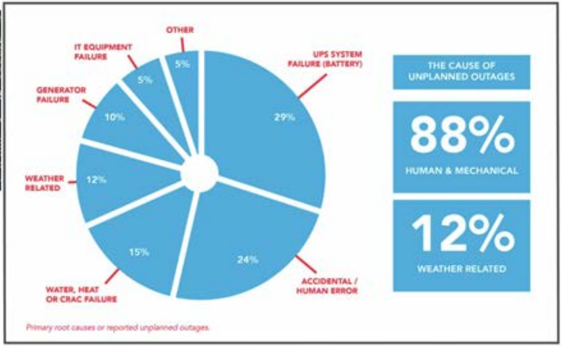
On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



On the necessity of adversary-resilient?

Security: external adversarial attacks, unstructured system failures, and consistent external disturbance



Remainder of this workshop

1. Byzantine consensus
   - Ensures secure and effective information fusion while using local communication only
2. Byzantine-resilient distributed optimization
   - Fundamental limits
   - Optimal algorithms
3. Byzantine-resilient light-weight social learning
   - The first provably secure algorithm
   - A light-weight variant

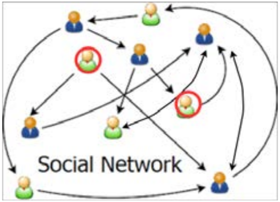
Remainder of this workshop

1. **Byzantine consensus**
   - Ensures secure and effective information fusion while using local communication only
2. Byzantine-resilient distributed optimization
   - Fundamental limits
   - Optimal algorithms
3. Byzantine-resilient light-weight social learning
   - The first provably secure algorithm
   - A light-weight variant

Byzantine Consensus

Communication network

- a collection of n agents communicating with each other through a network $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, n\}$ and $\mathcal{E}$ denote the set of agents and communication links, respectively.



- Among the n agents, an *unknown* subset of agents might be compromised and behave adversarially.

Fault/Adversary Model - I

Byzantine Fault Model: There exists a system adversary that can choose up to b out of n agents to compromise and control. Let $\mathcal{A} \subseteq \mathcal{N}$ be the set of compromised agents, referred to as *Byzantine agents*.

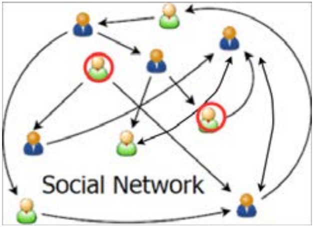
"The Byzantine Generals Problem", LAMPORT, SHOSTAK, and PEASE

- The adversary has complete knowledge of the network
  - the local program that each good agent is supposed to run;
  - the current status of the system;
  - running history of the system.

Fault/Adversary Model - II

The Byzantine agents can

- collude with each other;
- deviate from their pre-specified local programs to *arbitrarily* misrepresent information to the good agents;



- mislead each of the good agents in a unique fashion, i.e., letting $m_{ij}(t) \in \mathbb{R}^d$ be the message sent from agent $i \in \mathcal{A}$ to agent $j \in \mathcal{V} \setminus \mathcal{A}$ at time t, it is possible that $m_{ij}(t) \neq m_{ij'}(t)$ for $j \neq j' \in \mathcal{V} \setminus \mathcal{A}$.

Reaching agreement in the presence of Byzantine faults is far from trivial.

Example: For binary consensus, even in complete graphs, no distributed algorithms can tolerate more than 1/3 of the agents to be Byzantine. [Lamport, Shostak, and Pease, 82]

Reaching agreement in the presence of Byzantine faults is far from trivial.

Example: For binary consensus, even in complete graphs, no distributed algorithms can tolerate more than 1/3 of the agents to be Byzantine. [Lamport, Shostak, and Pease, 82]

The reached agreement could be biased and the amount of bias is out of the control of the good agents.

Background-I: Byzantine Fault-Tolerance

- proposed in [Pease–Shostak–Lamport, J. ACM80']

Background-I: Byzantine Fault-Tolerance

- proposed in [Pease–Shostak–Lamport, J. ACM80']
- FLP impossibility result: Asynchronous Byzantine consensus is impossible to solve (FLP impossibility) [Fischer – Lynch – Peterson, J. ACM85']

Background-I: Byzantine Fault-Tolerance

- proposed in [Pease–Shostak–Lamport, J. ACM80']
- FLP impossibility result: Asynchronous Byzantine consensus is impossible to solve (FLP impossibility) [Fischer – Lynch – Peterson, J. ACM85']
- Approximate Byzantine consensus: Relaxing the necessity of agree with each other **exactly** [Dolev et al., J. ACM86']
  - Initially proposed for asynchronous systems, extended to synchronous systems

Background-II

n: the total # of agents;
b: the maximal number of Byzantine (i.e., compromised) agents

- Communication with message relay:
  - Networks with bidirectional links [Fisher-Lynch-Paterson, PODC85']
    - $n \geq 3b + 1$, and $2b + 1$ node connectivity
  - Networks with directional links [Tseng-Vaidya, PODC15']
    - based on four sets nodes partition
- Local communication: an agent can only communicate with its immediate neighbors
  [Vaidya-Tseng-Liang, PODC'12], [LeBlanc et al., HiCoNS '12]

Questions Answered

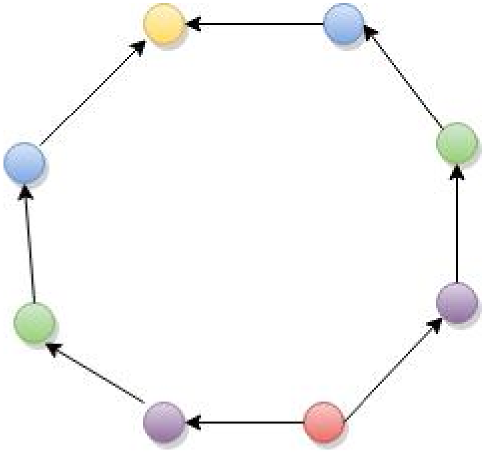
The impact of communication range:

- Will there be a tight topology condition over G?
- If yes, how does the communication range affect the tight condition?
- Is there any simple algorithm that works under the tight condition?

Model

- Synchronous system
- Communication network: *arbitrary directed* graph
  - Node i can send message to node j: if node j is reachable via at most ℓ hops.
  - A message is modeled as a tuple $m = (w, P)$.
  - Messages delivered by the network layer.
- Up to b Byzantine faults
  - Tamper messages value if it belongs to an admissible communication path, leaving message path unchanged.

Model



Approximate Consensus: Correctness Conditions

- ϵ–Agreement
- Validity: Outputs are within the range of inputs at fault-free nodes.
- Termination

Iterative Structure

Each fault-free node i maintains a state v_i: initial state = input

Algorithm Structure: For $t \geq 1$ and node i,

1. Transmit step.
2. Receive step. Let $\mathcal{M}_i[t]$ be the set of messages that node i in this step.
3. Update step: Node i updates its state as

$$v_i[t] = Z_i(\mathcal{M}_i[t]).$$

Iterative Structure

Each fault-free node i maintains a state v_i: initial state = input

Algorithm Structure: For $t \geq 1$ and node i,

1. Transmit step.
2. Receive step. Let $\mathcal{M}_i[t]$ be the set of messages that node i in this step.
3. Update step: Node i updates its state as

$$v_i[t] = Z_i(\mathcal{M}_i[t]).$$

- Minimal memory across iterations

Iterative Structure

Each fault-free node i maintains a state v_i: initial state = input

Algorithm Structure: For $t \geq 1$ and node i,

1. Transmit step.
2. Receive step. Let $\mathcal{M}_i[t]$ be the set of messages that node i in this step.
3. Update step: Node i updates its state as

$$v_i[t] = Z_i(\mathcal{M}_i[t]).$$

Question: Which directed graphs can solve *iterative* approximate Byzantine consensus?

Results

ℓ-restricted connectivity

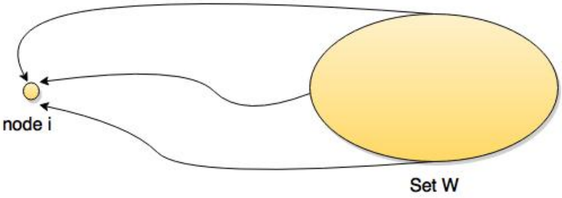
Definition (ℓ-restricted connectivity)

Suppose that $W \neq$ is a set of a node and that $x \notin W$. A node set S_ℓ with $x \notin S_\ell$ is called an ℓ-restricted (W, x) cut if the deletion of S_ℓ disconnects all (W, x)-paths of length up to ℓ. The ℓ-restricted (W, x) connectivity, denoted by $\kappa_\ell(W, x)$ is the size of the smallest ℓ-restricted (W, x) cut.

ℓ-restricted connectivity

Definition (ℓ-restricted connectivity)

Suppose that $W \neq$ is a set of a node and that $x \notin W$. A node set S_ℓ with $x \notin S_\ell$ is called an ℓ-restricted (W, x) cut if the deletion of S_ℓ disconnects all (W, x)-paths of length up to ℓ. The ℓ-restricted (W, x) connectivity, denoted by $\kappa_\ell(W, x)$ is the size of the smallest ℓ-restricted (W, x) cut.



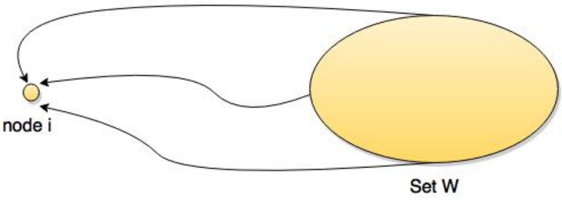
Node i is influenced by W if $\kappa_\ell(W, i) > b + 1$

Influence Relation

$\kappa_\ell(W, i) > b + 1 \quad \Longleftrightarrow$ node i is able to utilize at least one untampered message for its state update.

Influence Relation

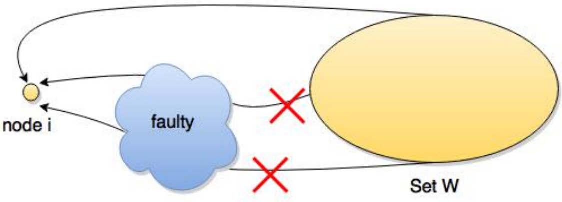
$\kappa_\ell(W, i) > b + 1 \quad \Longleftrightarrow$ node i is able to utilize at least one untampered message for its state update.



Node i is influenced by W

Influence Relation

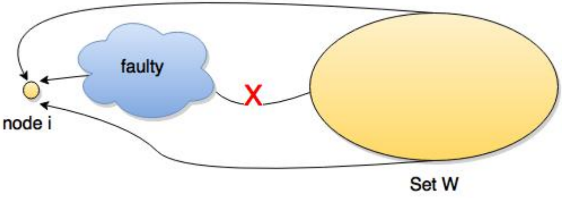
$\kappa_\ell(W, i) > b + 1 \iff$ node i is able to utilize at least one untampered message for its state update.



Node i is influenced by W

Influence Relation

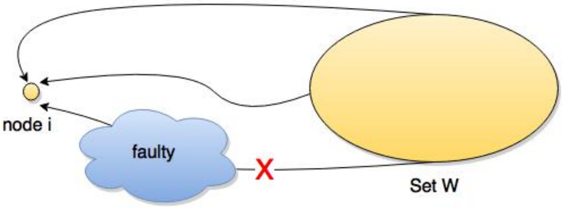
$\kappa_\ell(W, i) > b + 1 \quad \Longleftrightarrow$ node i is able to utilize at least one untampered message for its state update.



Node i is influenced by W

Influence Relation

$\kappa_\ell(W, i) > b + 1 \iff$ node i is able to utilize at least one untampered message for its state update.



Node i is influenced by W

Necessary Condition: Condition NC

Definition

For nonempty disjoint not sets A and B, we say $A \Rightarrow_\ell B$ if and only if there exists a node $i \in B$ such that $\kappa_\ell(A, i) \geq b + 1$.

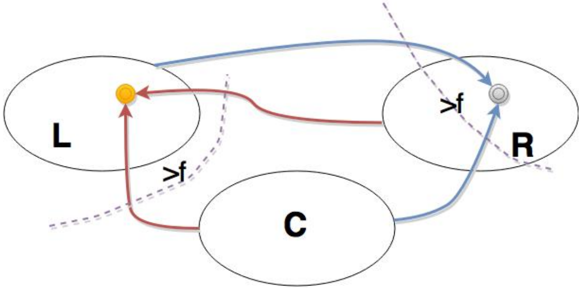
Condition NC for a given ℓ

For any node partition L, C, R, F of G such that $L \neq \emptyset$, $R \neq \emptyset$ and $|F| \leq b$, in G_F, at least one of the two conditions below must be true: (i) $R \cup C \Rightarrow_\ell L$; (ii) $L \cup C \Rightarrow_\ell R$.

Necessary Condition: Condition NC

Condition NC for a given ℓ

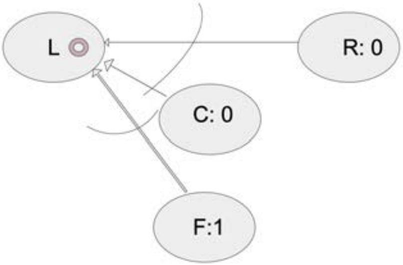
For any node partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq b$, in G_F, at least one of the two conditions below must be true: (i) $R \cup C \Rightarrow_\ell L$; (ii) $L \cup C \Rightarrow_\ell R$.



Either a golden or a silver node exists!

Necessary Condition: Proof Sketch

Suppose neither a golden nor a silver node exists. Suppose each node in L has value 1 and each node in R, and C has value 0. Byzantine nodes in F tell each node in L their values are all 1 and tell each node in R their values are 0.

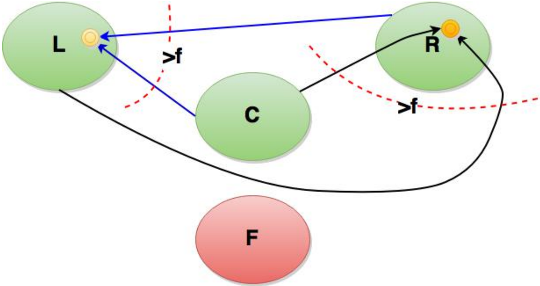


Each node in L does not know whether it should trust $R \cup C$ or F. If it chooses to trust $R \cup C$, then it should output 0. If it chooses to trust F, then it will update its value closer to 1.

Necessary Condition: Condition NC

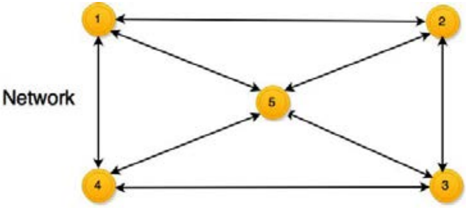
Condition NC for $\ell = 1$ [Vaidya-Tseng-Liang,PODC'12]

For any node partition L, C, R, F of G such that $L \neq \emptyset, R \neq \emptyset$ and $|F| \leq f$, in the induced subgraph G_F, at least one of the two conditions below must be true: (i) there exists a node $i \in L$ such that $\left|(R \cup C) \cap N_i^-\right| \geq b + 1$; (ii) there exists a node $j \in R$ such that $\left|(L \cup C) \cap N_j^-\right| \geq b + 1$.



Necessary Condition NC

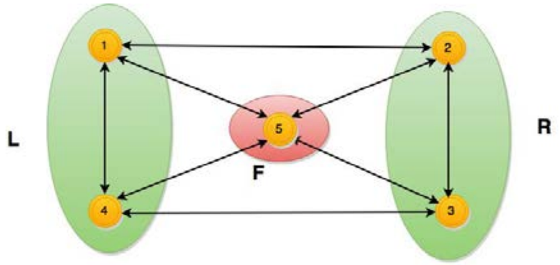
Allowing message relay (i.e., $\ell > 1$), the network necessary condition is strictly more relax than the one for single-hop message transmission model obtained in [Vaidya-Tseng-Liang, PODC12].



In this system, there are five nodes p_1, p_2, p_3, p_4 and p_5; all communication links are bi-directional; and at most one node can be adversarial, i.e., $b = 1$.

Necessary Condition NC

For $l > 1$, Condition NC is (in general) weaker than necessary condition derived under single-hop message transmission model obtained in [PODC12: Vaidya-Tseng-Liang].



- This graph does not satisfy the one in [PODC12: Vaidya-Tseng-Liang]
- satisfies our Condition NC for $\ell > 1$.

Recalling the iterative structure

Each fault-free node i maintains a state v_i: initial state = input

Algorithm Structure: For $t \geq 1$ and node i,

1. Transmit step.
2. Receive step. Let $\mathcal{M}_i[t]$ be the set of messages that node i in this step.
3. Update step: Node i updates its state as

$$v_i[t] = Z_i(\mathcal{M}_i[t]).$$

Recalling the iterative structure

Each fault-free node i maintains a state v_i: initial state = input

Algorithm Structure: For $t \geq 1$ and node i,

1. Transmit step.
2. Receive step. Let $\mathcal{M}_i[t]$ be the set of messages that node i in this step.
3. Update step: Node i updates its state as

$$v_i[t] = Z_i(\mathcal{M}_i[t]).$$

For $l = 1$, [PODC12: Vaidya et al.] and [HiCoNSa12: LeBlanc et al.] both use

"Adversarial Robust" update= trimming + averaging

Trimming Strategy

- **When** $\ell = 1$**:** remove extreme received message values – largest *f* values and smallest *f* values
- **When** $\ell > 1$**:** ?

Trimming Strategy

- **When** $\ell = 1$**:** remove extreme received message values – largest *f* values and smallest *f* values
- **When** $\ell > 1$**:** ?
  - message values

Trimming Strategy

- **When $\ell = 1$:** remove extreme received message values – largest *f* values and smallest *f* values
- **When $\ell > 1$:** ?
  - message values
  - message paths

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i, i))\}$.

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}_i'[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i,i))\}$.
- Sort messages in $\mathcal{M}_i'[t]$ in an increasing order.

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i,i))\}$.
- Sort messages in $\mathcal{M}'_i[t]$ in an increasing order.
- Let $\mathcal{M}_{is}[t]$ be the largest sized subset of $\mathcal{M}'_i[t]$ such that
  (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{is}[t]$ and $m' \in \mathcal{M}_{is}[t]$ we have value$(m) \geq$ value(m'),
  (ii) at least f nodes are needed to collectively tamper all messages in $\mathcal{M}_{is}[t]$.

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i, i))\}$.
- Sort messages in $\mathcal{M}'_i[t]$ in an increasing order.
- Let $\mathcal{M}_{is}[t]$ be the largest sized subset of $\mathcal{M}'_i[t]$ such that
  - (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{is}[t]$ and $m' \in \mathcal{M}_{is}[t]$ we have value$(m) \geq$ value(m'),
  - (ii) at least f nodes are needed to collectively tamper all messages in $\mathcal{M}_{is}[t]$.
- Set $\mathcal{M}_{il}[t]$ is constructed similarly.

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i, i))\}$.
- Sort messages in $\mathcal{M}'_i[t]$ in an increasing order.
- Let $\mathcal{M}_{is}[t]$ be the largest sized subset of $\mathcal{M}'_i[t]$ such that
  (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{is}[t]$ and $m' \in \mathcal{M}_{is}[t]$ we have $\text{value}(m) \geq \text{value}(m')$,
  (ii) at least f nodes are needed to collectively tamper all messages in $\mathcal{M}_{is}[t]$.
- Set $\mathcal{M}_{il}[t]$ is constructed similarly.

Trimming Strategy: Removed Messages Set Construction

For each i, the trimmed messages sets $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are constructed (identified) as

- Let $\mathcal{M}'_i[t] = \mathcal{M}_i[t] - \{(v_i[t-1], (i,i))\}$.
- Sort messages in $\mathcal{M}'_i[t]$ in an increasing order.
- Let $\mathcal{M}_{is}[t]$ be the <u>largest sized</u> subset of $\mathcal{M}'_i[t]$ such that
  - (i) for all $m \in \mathcal{M}'_i[t] - \mathcal{M}_{is}[t]$ and $m' \in \mathcal{M}_{is}[t]$ we have $\text{value}(m) \geq \text{value}(m')$,
  - (ii) at least f nodes are needed to collectively tamper all messages in $\mathcal{M}_{is}[t]$.
- Set $\mathcal{M}_{il}[t]$ is constructed similarly.

Both $\mathcal{M}_{is}[t]$ and $\mathcal{M}_{il}[t]$ are well-defined.

Algorithm 1

1. *Transmit step.*
2. *Receive step.*
3. *Update step:*

$$v_i[t] = \frac{1}{|\mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]|} \sum_{m \in \mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]} w_m$$

Proof of Correctness

$v_i[t]$: state of fault-free node i at the end of iteration t
$\mathbf{v}[t]$: vector of states of fault-free nodes

Proof ideas

- Construct a proper matrix $\mathbf{M}[t]$ such that

$$\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1].$$

Proof of Correctness

$v_i[t]$: state of fault-free node i at the end of iteration t
$\mathbf{v}[t]$: vector of states of fault-free nodes

Proof ideas

- Construct a proper matrix $\mathbf{M}[t]$ such that

$$\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1].$$

- Then

$$\mathbf{v}[t] = (\mathbf{M}[t]\mathbf{M}[t-1]\cdots\mathbf{M}[0])\,\mathbf{v}[0]$$

Proof of Correctness

$v_i[t]$: state of fault-free node i at the end of iteration t
$\mathbf{v}[t]$: vector of states of fault-free nodes

Proof ideas

- Construct a proper matrix $\mathbf{M}[t]$ such that

$$\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1].$$

- Then

$$\mathbf{v}[t] = \left(\mathbf{M}[t]\mathbf{M}[t-1]\cdots\mathbf{M}[0]\right)\mathbf{v}[0]$$

- When $G(\mathcal{V}, \mathcal{E})$ satisfies Condition NC,

$$\lim_t \mathbf{M}[t]\mathbf{M}[t-1]\cdots\mathbf{M}[0] = \mathbf{M}^* = \mathbf{1}\cdot\pi^T.$$

Matrix Construction

Recall that

$$v_i[t] \;=\; \frac{1}{|\mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]|} \sum_{m \in \mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]} w_m \qquad (1)$$

To go from (1) to

$$\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1]$$

...

- Messages are collected over the G^ℓ
- Update graph is a subgraph of $(G_F)^\ell$
- Weights reallocation

Matrix Construction

Recall that

$$v_i[t] = \frac{1}{|\mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]|} \sum_{m \in \mathcal{M}_i[t] - \mathcal{M}_{is}[t] - \mathcal{M}_{il}[t]} w_m \qquad (1)$$

To go from (1) to

$$\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1]$$

...

- Messages are collected over the G^ℓ
- Update graph is a subgraph of $(G_F)^\ell$
- Weights reallocation

Condition NC guarantees that there exists a unique source component in the update graph.

Connection with existing work under unbounded path length

When G is undirected [Fischer-Lynch-Merritt,PODC85]

Theorem (Undirected Graph)

When $\ell \geq \ell^*$, if G is undirected, then $n \geq 3b + 1$ and node-connectivity of G is at least $2b + 1$ if and only if G satisfies Condition NC.

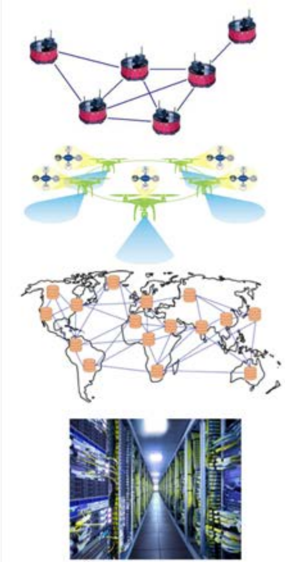
Connection with existing work under unbounded path length

When G is directed [PODC15: Tseng-Vaidya]
$B \rightarrow A$: Set A is influenced by set B if

- $A \cap B = \emptyset$
- nodes in A collectively have at least $b + 1$ distinct incoming neighbors in B

Fault-Tolerant Distributed Optimization in Multi-Agent Networks

System Goal: Secure Multi-Agent Optimization



Cooperatively optimizing a global objective through inter-agent communication and local computations in the presence of faulty agents

Examples

- Robotic rendezvous problems.
- Parameter estimation in distributed sensor networks:
  - Regression-based estimates using local sensor measurements
- Large-scale distributed machine learning, where data are generated at different locations

Outline

- Review: Faulty free
- Crash failure and Byzantine-resilience
- Impossibility results for Byzantine-resilience
- Algorithms for Byzantine-resilience
- Optimization problem with additional structures

Outline

- **Review: Faulty free**
- Crash failure and Byzantine-resilience
- Impossibility results for Byzantine-resilience
- Algorithms for Byzantine-resilience
- Optimization problem with additional structures

Model

- We consider a network of n agents with node set $\mathcal{V} = [1, 2, \ldots, n]$.
- Each agent i locally has its own convex objective function $h_i(x) : \mathbb{R} \to \mathbb{R}$.

Goal (Failure-Free)

Agents want to cooperatively minimize

$$h(x) = \frac{1}{n} \sum_{i=1}^{n} h_i(x).$$

[Nedic and Ozdaglar, 2009], [Duchi et al., 2012], [Tsianos et al., 2012]

Examples

- Robotic rendezvous:
  - $h_i(x)$: agent i's cost for rendezvous.
  - $h(x)$: cost for rendezvous.
- Parameter estimation in distributed sensor networks:
  - Regression-based estimates using local sensor measurements
- Large-scale distributed machine learning, where data are generated at different locations

Example: Empirical Risk Minimization

Suppose data is collected by different agents

- agent j keeps local data $\{x_{j_i}, y_{j_i}\}_{i=1}^{m_j}$, $j = 1, \cdots, n$
- Loss function: L, with $L(x_{j_i}, y_{j_i}, \theta)$
- Without communication: Locally minimizing $f_j(\theta) := \sum_{i=1}^{m_j} L(x_{j_i}, y_{j_i}, \theta)$
- With communication: Globally solving ([Nedic and Ozdaglar, 2009], [Duchi et al. 2012], and etc.)

$$\min_{\theta} \quad \frac{1}{n} \sum_{j=1}^{n} f_j(\theta) = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{m_j} L(x_i, y_i, \theta)$$

Algorithm (fault-free) [Nedic and Ozdaglar, 2009]

- Compute $h_i'(x_i[t])$;
- Send $x_i[t]$ to nodes in N_i^+ – the outgoing neighbors of i;
- Receive $x_j[t]$ from all its incoming neighbors N_i^-;
-

$$x_i[t+1] \leftarrow \frac{1}{|N_i^-|+1}\left(\sum_{j \in N_i^- \cup \{i\}} x_i[t]\right) - \lambda[t] h_i'(x_i[t])$$

Algorithm (fault-free) [Nedic and Ozdaglar, 2009]

- Compute $h_i'(x_i[t])$;
- Send $x_i[t]$ to nodes in N_i^+ – the outgoing neighbors of i;
- Receive $x_j[t]$ from all its incoming neighbors N_i^-;
- $$x_i[t+1] \leftarrow \frac{1}{|N_i^-|+1}\left(\sum_{j \in N_i^- \cup \{i\}} x_j[t]\right) - \lambda[t] h_i'(x_i[t])$$

  $$= x_i[t] - \lambda[t] h_i'(x_i[t]) + \frac{1}{|N_i^-|+1} \sum_{j \in N_i^-} (x_j[t] - x_i[t])$$

It can be shown that for sufficient large t, we have for each $i \in \mathcal{V}$

$$x_i[t+1] \approx x_i[t] - \lambda[t] \frac{1}{n} \sum_{i=1}^{n} h_i'(x_i[t]),$$

Outline

- Review: Faulty free
- **Crash failure and Byzantine-resilience**
- Impossibility results for Byzantine-resilience
- Algorithms for Byzantine-resilience
- Optimization problem with additional structures

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

When $f > 0$, it is impossible to solve $h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$.

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

When $f > 0$, it is impossible to solve $h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$.

Question

What should be the global objectives?

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

When $f > 0$, it is impossible to solve $h(x) = \frac{1}{n} \sum_{i=1}^{n} h_i(x)$.

Question

What should be the global objectives?

Observations:

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

When $f > 0$, it is impossible to solve $h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$.

Question

What should be the global objectives?

Observations:

1. Only available and untampered h_i should be used.

Fault-Tolerant Multi-Agent Optimization

- Fault models: Crash and Byzantine faults
- System models: Synchronous and asynchronous systems

When $f > 0$, it is impossible to solve $h(x) = \frac{1}{n}\sum_{i=1}^{n} h_i(x)$.

Question

What should be the global objectives?

Observations:

1. Only available and untampered h_i should be used.
2. Sufficient number of h_i's should be used.

Assumptions on Local cost functions

- $h_i : \mathbb{R} \to \mathbb{R}$
- convex, and continuously differentiable
- optimal set is non-empty and compact (i.e., bounded and closed)
- bounded gradient
- L–Lipschitz gradients

Global Objective: Crash Resilience – I

Up to f agents may crash – their local functions unavailable

Goal ($f > 0$, crash fault)

Non-faulty agents want to collaboratively minimize an unknown function of the form

$$h(x) = \sum_{i \in \mathcal{V}} \alpha_i h_i(x),$$

where $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i = 1$, and depend on the failure pattern of the faulty agents.

When $\mathcal{F} = \{1, \ldots, f\}$ and crash at time $t = 0$, it holds that $\alpha_i = 0$ for $i = 1, \ldots, f$.

Intuitively speaking, the coefficients α_i's capture the utilization level of individual measurements.

Quality of the Output

- Only convex combination: multiple output candidates
- How to measure the quality of an output candidate?

Quality of the Output

- Only convex combination: multiple output candidates
- How to measure the quality of an output candidate?

(β, γ)–admissibility of a given α ($\beta > 0$, and $\gamma \in$):

At least γ elements of α are lower bounded by β

not $(\frac{2}{10}, 4)$–admissible

Quality of the Output

- Only convex combination: multiple output candidates
- How to measure the quality of an output candidate?

(β, γ)–admissibility of a given α ($\beta > 0$, and $\gamma \in$):

At least γ elements of α are lower bounded by β

Example: $\alpha = \{\frac{1}{10}, \frac{3}{10}, 0, 0, \frac{4}{10}, \frac{2}{10}, 0\}$

is $(\frac{1}{10}, 4)$–admissible
not $(\frac{2}{10}, 4)$–admissible

Global Objective: Crash Resilience – II

Introducing two parameters $\beta \geq 0$ and $\gamma \geq 0$.

Non-faulty agents aim to minimize an unknown function

$$h(x) = \sum_{i \in \mathcal{V}} \alpha_i h_i(x),$$

such that

$$\forall i \in \mathcal{V},\ \alpha_i \geq 0,\ \sum_{i \in \mathcal{V}} \alpha_i = 1,$$

$$\text{and} \quad \sum_{i \in \mathcal{V}} \mathbf{1}(\alpha_i \geq \beta) \geq \gamma.$$

[Su and Vaidya,arxiv'15c]

1. Synchronous system: $\alpha_i = \alpha_j \geq \frac{1}{n}$ for all $i, j \in \mathcal{N}$.
2. Asynchronous system: $\alpha_i \geq \frac{1}{n}$ for all $i \in \mathcal{N}$.

Global Objective: Byzantine Resilience

Up to f agents may be Byzantine – they can hide and adaptively lie about their local functions

Refined Goal ($f > 0$, Byzantine fault) for $\beta \geq 0$ and $\gamma \geq 0$

Non-faulty agents want to collaboratively minimize an unknown function of the form

$$h(x) = \sum_{i \in \mathcal{N}} \alpha_i h_i(x),$$

such that

$$\forall i \in \mathcal{N},\ \alpha_i \geq 0,\ \sum_{i \in \mathcal{N}} \alpha_i = 1,$$

$$\text{and} \quad \sum_{i \in \mathcal{N}} \mathbf{1}(\alpha_i \geq \beta) \geq \gamma.$$

Henceforth, we consider synchronous system.

Outline

- Review: Faulty free
- Crash failure and Byzantine-resilience
- **Impossibility results for Byzantine failure**
- Algorithms for Byzantine-resilience
- Optimization problem with additional structures

Impossibility Results

Theorem 1 [S. and Vaidya, TAC'20]

When $f > 0$, it is impossible to minimize

$$h(x) = \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|} h_i(x).$$

Impossibility Results

Theorem 1 [S. and Vaidya, TAC'20]

When $f > 0$, it is impossible to minimize

$$h(x) = \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|} h_i(x).$$

Intuition: Need to identify which agents are Byzantine.
Impossible under data heterogeneity!!!

Impossibility Results

Theorem 1 [S. and Vaidya, TAC'20]

When $f > 0$, it is impossible to minimize

$$h(x) = \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|} h_i(x).$$

Intuition: Need to identify which agents are Byzantine.
Impossible under data heterogeneity!!!

Theorem 2 [S. and Vaidya, TAC'20]

It is impossible to achieve $\beta \geq \epsilon$ and $\gamma > |\mathcal{N}| - f$ regardless of the choice of $\epsilon > 0$.

Impossibility Results

Theorem 1 [S. and Vaidya, TAC'20]

When $f > 0$, it is impossible to minimize

$$h(x) = \sum_{i \in \mathcal{N}} \frac{1}{|\mathcal{N}|} h_i(x).$$

Intuition: Need to identify which agents are Byzantine.
Impossible under data heterogeneity!!!

Theorem 2 [S. and Vaidya, TAC'20]

It is impossible to achieve $\beta \geq \epsilon$ and $\gamma > |\mathcal{N}| - f$ regardless of the choice of $\epsilon > 0$.

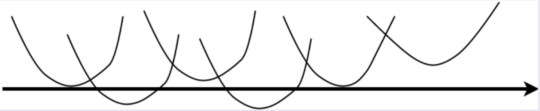
Remark: Byzantine resilience comes at a price of sacrificing the information collected by at least f non-faulty agents

Outline

- Review: Faulty free
- Crash failure and Byzantine-resilience
- Impossibility results for Byzantine failure
- **Algorithms for Byzantine-resilience**
- Optimization problem with additional structures

Algorithm 1: Broadcasting local functions

Step 1: Perform Byzantine broadcast for each of $h_j(x)$.



The n local functions collected by agent j

Step 2: If there exists $x_0 \in \mathbb{R}$ such that

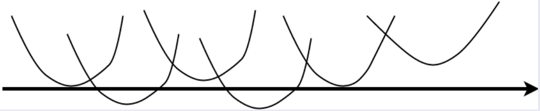
$$\sum_{i \in A(x) - F_1^*(x)} h_i'(x) + \sum_{i \in B(x) - F_2^*(x)} h_i'(x) \;=\; 0$$

then output $\widetilde{x} = x_0$; otherwise, output $\widetilde{x} = \perp$.

On $F_1^*(x)$ **and** $F_2^*(x)$: F_1^* largest f (if any) positive gradient, F_2^* smallest f (if any) negative gradient

Algorithm 1: Broadcasting local functions

Step 1: Perform Byzantine broadcast for each of $h_j(x)$.



The n local functions collected by agent j

Step 2: If there exists $x_0 \in \mathbb{R}$ such that

$$\sum_{i \in A(x) - F_1^*(x)} h_i'(x) + \sum_{i \in B(x) - F_2^*(x)} h_i'(x) \quad = \quad 0$$

then output $\widetilde{x} = x_0$; otherwise, output $\widetilde{x} = \perp$.

Theorem 3[S. and Vaidya, TAC'20, arXiv 2015'a]

When $n > 3f$, Algorithm 1 achieves the refined goal with $\beta = \max\{\frac{1}{n}, \frac{1}{2(|\mathcal{N}| - f)}\}$ and $\gamma = |\mathcal{N}| - f$.

Algorithm 1: Alternative Interpretation

For each $x \in \mathbb{R}$, let

$$H(x) = \sum_{i \in A(x) - F_1^*(x)} h_i'(x) + \sum_{i \in B(x) - F_2^*(x)} h_i'(x).$$

Theorem

For given $\mathcal{N}$ and $\mathcal{F}$, there exists a convex and differentiable function $\mathbf{H}(\cdot)$ such that $\mathbf{H}'(x) = H(x)$.

Essentially, the above algorithm outputs an optimum of the following constrained convex optimization problem, where $Cov\left(\cup_{i \in \mathcal{N}} X_i\right) \subseteq [c, d]$:

$$\begin{aligned} \min \quad & \mathbf{H}(x) \\ s.t. \quad & x \in [c, d]. \end{aligned}$$

Algorithm 2: Gradient Broadcast + Admissibility Check

Algorithm 2: Agent j for $j \in \mathcal{N}$

- Perform Byzantine consensus on initial estimates $x_j[0]$'s.
- Compute $h_j'(x_j[t])$, and perform Byzantine broadcast of $h_j'(x_j[t])$ to all the agents.
- Admissibility check on received gradients $\{g_1[t], \ldots, g_n[t]\}$.
- Trim away extreme gradients. Let $\mathcal{R}_j^*[t]$ be the agents whose gradients have not been removed.
  - $x_j[t+1] \leftarrow x_j[t] - \lambda[t] \sum_{i \in \mathcal{R}^*[t]} g_i[t]$.

Algorithm 2: Gradient Broadcast + Admissibility Check

Algorithm 2: Agent j for $j \in \mathcal{N}$

- Perform Byzantine consensus on initial estimates $x_j[0]$'s.
- Compute $h_j'(x_j[t])$, and perform Byzantine broadcast of $h_j'(x_j[t])$ to all the agents.
- Admissibility check on received gradients $\{g_1[t], \ldots, g_n[t]\}$.
- Trim away extreme gradients. Let $\mathcal{R}_j^*[t]$ be the agents whose gradients have not been removed.
  - $x_j[t+1] \leftarrow x_j[t] - \lambda[t] \sum_{i \in \mathcal{R}^*[t]} g_i[t]$.

Admissibility check: check whether the received gradients can be interpreted as the gradient of some convex functions.

Diminishing stepsizes: $\lambda[t] \to 0$, $\sum_{t=0}^{\infty} \lambda[t] = \infty$ and $\sum_{t=0}^{\infty} \lambda^2[t] < \infty$.

Correctness of Algorithm 2: Proof Ideas

1. Identical estimates at non-faulty agents: $x_j[t] = x_i[t]$, for $i, j \in \mathcal{N}$. Let $x[t] = x_j[t]$.
2. Admissibility check force the faulty agents behave as if its local function is admissible. Thus agent i keeps local function $h_i(\cdot)$ for each $i \in \mathcal{V}$.
3. Let $H(\cdot)$ be defined as before, i.e.,

$$H(x) = \sum_{i \in A(x) - F_1^*(x)} h_i'(x) + \sum_{i \in B(x) - F_2^*(x)} h_i'(x)\,.$$

4. Indeed,

$$\begin{aligned} x[t+1] &= x[t] - \lambda[t] \sum_{i \in \mathcal{R}^*[t]} g_i[t] \\ &= x[t] - \lambda[t] H(x[t]). \end{aligned}$$

Algorithm 2: Alternative Interpretation

Since

$$
\begin{aligned}
x[t+1] &= x[t] - \lambda[t] \sum_{i \in \mathcal{R}^*[t]} g_i[t] \\
&= x[t] - \lambda[t] H(x[t]),
\end{aligned}
$$

The agents in the network are collaboratively solving

$$
\begin{array}{ll}
\min & \mathbf{H}(x) \\
s.t. & x \in [c, d],
\end{array}
$$

using gradient descent method.

So far ...

- Byzantine broadcast communication is costly.
- Fully distributed algorithm exists, in which only local communication and local computation is needed.
  - In particular, at each iteration t, agent j computes its local gradient at $x_j[t]$ and sends both $x_j[t]$ and its gradient to the other agents.
  - Trim over received estimates $x_i[t]$'s and over received gradients, respectively.

Algorithm 3: An Optimal Fully Distributed Algorithm

Theorem (S. and Vaidya, PODC'16)

There exists a distributed algorithm whose output admits an α that is (β, γ)–admissible with $\gamma = |\mathcal{N}| - f$ and $\beta = \frac{1}{2(|\mathcal{N}|-f)}$.

Algorithm 3: An Optimal Fully Distributed Algorithm

Theorem (S. and Vaidya,PODC'16)

There exists a distributed algorithm whose output admits an α that is (β, γ)–admissible with $\gamma = |\mathcal{N}| - f$ and $\beta = \frac{1}{2(|\mathcal{N}|-f)}$.

- $\gamma = |\mathcal{N}| - f$ is optimal [S. and Vaidya,16 ACC]
- $\beta = \frac{1}{2(|\mathcal{N}|-f)}$ is "off" by a factor of 2

  – observing that the largest possible β is $\frac{1}{|\mathcal{N}|-f}$
- Communication network: complete graph
- Can be extended to incomplete graphs [S.and Vaidya,arXiv'15d]

Assumptions

- Local cost functions
  - $h_i : \mathbb{R} \to \mathbb{R}$
  - convex, and continuously differentiable
  - optimal set is non-empty and compact (i.e., bounded and closed)
  - bounded gradient
  - L–Lipschitz gradients

SBG: Synchronous Byzantine Gradient Method

gradient descent method + iterative Byzantine approximate consensus

Each agent i maintains local estimate $x_i[t]$

SBG (In each iteration)

- Send estimate $x_i[t]$ and gradient $h_i'(x_i[t])$ to all agents;
- $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \times \text{Trim}\{h'[t]\}$

SBG: Synchronous Byzantine Gradient Method

gradient descent method + iterative Byzantine approximate consensus

Each agent i maintains local estimate $x_i[t]$

SBG (In each iteration)

- Send estimate $x_i[t]$ and gradient $h_i'(x_i[t])$ to all agents;
- $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \times \text{Trim}\{h'[t]\}$

Trim: drop smallest f, largest f values, and take the mean of the remained values

SBG: Synchronous Byzantine Gradient Method

gradient descent method + iterative Byzantine approximate consensus

Each agent i maintains local estimate $x_i[t]$

SBG (In each iteration)

- Send estimate $x_i[t]$ and gradient $h_i'(x_i[t])$ to all agents;
- $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \times \text{Trim}\{h'[t]\}$

Trim: drop smallest f, largest f values, and take the mean of the remained values

Trim gradients: impose structure

Proof Outline

i **Consensus**:

$$\lim_{t} \left(x_i[t] - x_j[t]\right) = 0, \quad \text{for all } i, j \in \mathcal{N}$$

ii **Optimality**:

$$x_i[t] \text{ is asymptotically } \left(\frac{1}{2(|\mathcal{N}| - f)}, |\mathcal{N}| - f\right)\text{–admissible}$$

Asymptotically $x_i[t]$ minimizes $\sum_{j\in\mathcal{N}} \alpha_j h_j(x)$ such that
α is (β, γ)–admissible with $\gamma = |\mathcal{N}| - f$ and $\beta = \frac{1}{2(|\mathcal{N}|-f)}$.

Characterization of Desired Outputs

Valid function p:

- $p(x) = \sum_{i \in \mathcal{N}} \alpha_i h_i(x)$
- weight vector α is $(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f)$–admissible

Characterization of Desired Outputs

Valid function p:

- $p(x) = \sum_{i \in \mathcal{N}} \alpha_i h_i(x)$
- weight vector α is $(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f)$–admissible

Set Y: all minimizers of valid functions

Characterization of Desired Outputs

Valid function p:

- $p(x) = \sum_{i \in \mathcal{N}} \alpha_i h_i(x)$
- weight vector α is $(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f)$–admissible

Set Y: all minimizers of valid functions

Lemma

Set Y is convex and closed.

Characterization of Desired Outputs

Valid function p:

- $p(x) = \sum_{i \in \mathcal{N}} \alpha_i h_i(x)$
- weight vector α is $(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f)$–admissible

Set Y: all minimizers of valid functions

Lemma

Set Y is convex and closed.

Optimality:

$x_i[t]$ is asymptotically $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right)$–admissible

$$\iff \lim_t \mathit{Distance}\,(x_i[t], Y) = 0$$

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

Lemma

Trim$\{h'[t]\}$ *at agent* i $= \sum_{j \in \mathcal{N}} \alpha_j^i[t] h_j'(x_j[t])$,

where $\alpha^i[t]$ *is* $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right)$ *–admissible*

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

Lemma

Trim$\{h'[t]\}$ at agent i $= \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t])$,

where $\alpha^i[t]$ is $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$–admissible

$$x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t])$$

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

Lemma

Trim$\{h'[t]\}$ *at agent* $i = \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t])$,

where $\alpha^i[t]$ *is* $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$ *–admissible*

$$x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t])$$

$$\approx x_i[t] - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_i[t])$$

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

Lemma

Trim$\{h'[t]\}$ *at agent* $i = \sum_{j\in\mathcal{N}} \alpha^i_j[t] h'_j(x_j[t])$,

where $\alpha^i[t]$ *is* $\left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}|-f\right)$ *–admissible*

$$\begin{aligned} x_i[t+1] &= \text{Trim}\{x[t]\} - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha^i_j[t] h'_j(x_j[t]) \\ &\approx x_i[t] - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha^i_j[t] h'_j(x_i[t]) \end{aligned}$$

$\sum_{j\in\mathcal{N}} \alpha^i_j[t] h'_j(x_i[t])$: the gradient of a valid function $p_{t,i}$

A Variant of Gradient Decent Update Rule

Update rule: $x_i[t+1] = \text{Trim}\{x[t]\} - \lambda[t] \cdot \text{Trim}\{h'[t]\}$

Lemma

$$\textit{Trim}\{h'[t]\} \textit{ at agent } i = \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t]),$$

$$\textit{where } \alpha^i[t] \textit{ is } \left(\frac{1}{2(|\mathcal{N}|-f)}, |\mathcal{N}| - f\right) \textit{–admissible}$$

$$\begin{aligned} x_i[t+1] &= \text{Trim}\{x[t]\} - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_j[t]) \\ &\approx x_i[t] - \lambda[t] \cdot \sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_i[t]) \\ &= x_i[t] - \lambda[t] \cdot p_{t,i}'(x_i[t]) \end{aligned}$$

$\sum_{j\in\mathcal{N}} \alpha_j^i[t] h_j'(x_i[t])$: the gradient of a valid function $p_{t,i}$

Remaining Optimality Proof

"gradient descent analysis" on the auxiliary $\{z[t]\}_{t=0}^{\infty}$ such that

$$z[t] = x_{j_t}[t]$$

$$\text{where } j_t \in_{j \in \mathcal{N}} \textit{Distance}\left(x_j[t], Y\right).$$

Remaining Optimality Proof

“gradient descent analysis" on the auxiliary $\{z[t]\}_{t=0}^{\infty}$ such that

$$z[t] = x_{j_t}[t]$$
$$\text{where } j_t \in_{j \in \mathcal{N}} \textit{Distance}\left(x_j[t], Y\right).$$

Intuitions behind optimality:

- The gradient of any valid function pushes $x_i[t]$ towards Y
- Since Y is convex, $x_i[t]$ is asymptotically trapped in Y

Open Problems

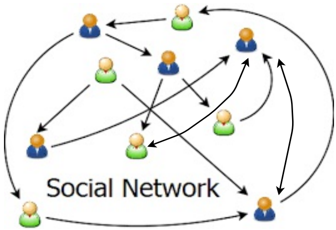
- General local function: vector inputs
  β, γ scale poorly in the input dimension *d*
- Incomplete graphs [S. and Vaidya, TAC'20]: weights might not be optimal

Open Problems

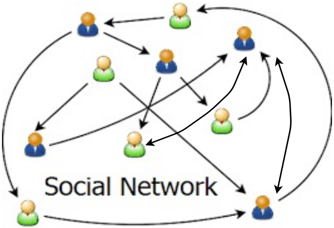
- General local function: vector inputs
  β, γ scale poorly in the input dimension *d*
- Incomplete graphs [S. and Vaidya, TAC'20]: weights might not be optimal

What if we have additional structures?

Learning in Multi-Agent Networks

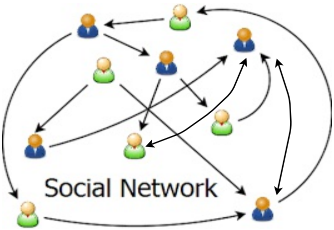


Learning in Multi-Agent Networks



- Each agent makes local observations
- Communicate with others

Learning in Multi-Agent Networks



- Each agent makes local observations
- Communicate with others

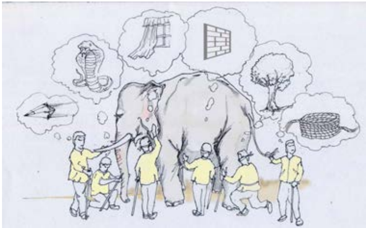
Who should be the President? Biden, Trump
What is the object in the sky? Bird, Plane, Missile, Meteor

Outline

- Review: Faulty free
- Crash failure and Byzantine-resilience
- Impossibility results for Byzantine-resilience
- Algorithms for Byzantine-resilience
- **Optimization problem with additional structures**

Learning over Multi-Agent Network (contd)

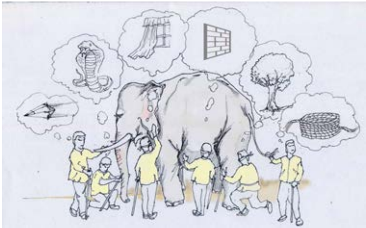
- Local observations: partially informative



$\boxed{\text{Collaboration is necessary!}}$

Learning over Multi-Agent Network (contd)

- Local observations: partially informative

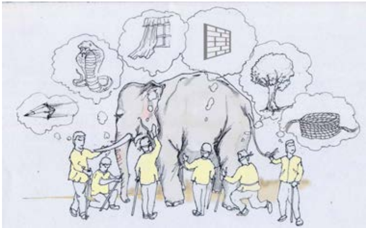


$\boxed{\text{Collaboration is necessary!}}$

- Some agents are adversarial: prevent the truth being learned

Learning over Multi-Agent Network (contd)

- Local observations: partially informative



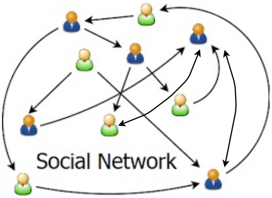
Collaboration is necessary!

- Some agents are adversarial: prevent the truth being learned

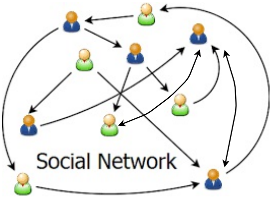
Goal: Non-faulty agents collaboratively learn the truth

Problem Formulation

- n agents in a directed network
- θ^*: unknown true state
- m possible states: $\theta_1, \ldots, \theta_m$



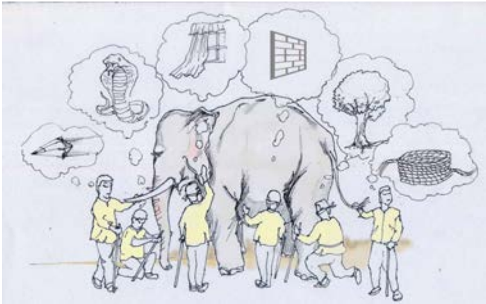
Problem Formulation



- n agents in a directed network
- θ^*: unknown true state
- m possible states: $\theta_1, \ldots, \theta_m$
- $s_t^i \sim \ell_{\theta^*}^i$: private signals of agent i at time t

Problem Formulation (contd)

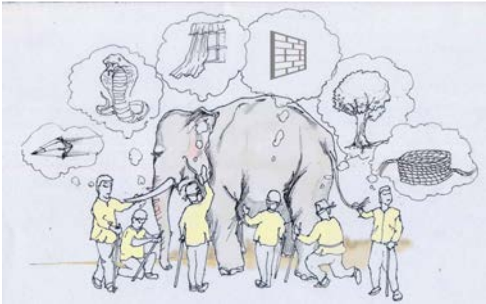
- Local indistinguishability



- Byzantine fault model: Up to f agents suffering Byzantine faults

Problem Formulation (contd)

- Local indistinguishability



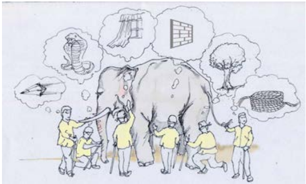
- Byzantine fault model: Up to f agents suffering Byzantine faults

Goal: Non-faulty agents collaboratively learn θ^*

Local Information

KL divergence $D_{KL}(\ell^i_{\theta_j} \parallel \ell^i_{\theta_k}) = 0$ iff the two distributions identical

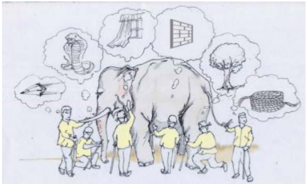
$\Rightarrow$ θ_j and θ_k indistinguishable to agent i



Local Information

KL divergence $D_{KL}(\ell^i_{\theta_j} \parallel \ell^i_{\theta_k}) = 0$ iff the two distributions identical

$\Rightarrow$ θ_j and θ_k indistinguishable to agent i



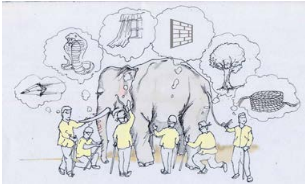
- $D_{KL}(l^i_{elephant} \parallel l^i_{tree}) = 0$

  $\Rightarrow$ *elephant* and *tree* look alike to agent i

Local Information

KL divergence $D_{KL}(\ell^i_{\theta_j} \parallel \ell^i_{\theta_k}) = 0$ iff the two distributions identical

$\Rightarrow$ θ_j and θ_k indistinguishable to agent i



- $D_{KL}(l^i_{elephant} \parallel l^i_{tree}) = 0$

  $\Rightarrow$ *elephant* and *tree* look alike to agent i

- $D_{KL}(l^i_{elephant} \parallel l^i_{tree}) > 0$

  $\Rightarrow$ *elephant* not confused with a *tree* by agent i

Global Information

$\sum_i D_{KL}(\ell^i_{\theta^*} \parallel \ell^i_\theta) \neq 0$ for all $\theta \neq \theta^*$

$\Rightarrow$ Collectively agents can distinguish θ^* (elephant) from $\theta \neq \theta^*$

Global Information

$\sum_i D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta) \neq 0$ for all $\theta \neq \theta^*$

$\Rightarrow$ Collectively agents can distinguish θ^* (elephant) from $\theta \neq \theta^*$

- When all agents cooperate, this suffices to learn θ^*

Global Information

$\sum_i D_{KL}(\ell^i_{\theta^*} \parallel \ell^i_\theta) \neq 0 \quad$ for all $\theta \neq \theta^*$

$\Rightarrow$ Collectively agents can distinguish θ^* (elephant) from $\theta \neq \theta^*$

- When all agents cooperate, this suffices to learn θ^*
- Not sufficient with Byzantine agents

Our Contributions [S. and Vaidya, DC'18]

1. Sufficient condition on l_θ^i's for learning with Byzantine faults

Our Contributions [S. and Vaidya, DC'18]

1. Sufficient condition on l_θ^i's for learning with Byzantine faults
2. First distributed learning algorithm robust to Byzantine faults

Our Contributions [S. and Vaidya, DC'18]

1. Sufficient condition on l_θ^i's for learning with Byzantine faults
2. First distributed learning algorithm robust to Byzantine faults
3. Proved fast convergence of the proposed algorithm

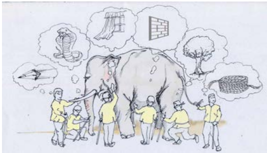
Our Contributions [S. and Vaidya, DC'18]

1. Sufficient condition on l_θ^i's for learning with Byzantine faults
2. First distributed learning algorithm robust to Byzantine faults
3. Proved fast convergence of the proposed algorithm
4. *Recent results:* A light-weight algorithm

Local Information v.s. Global Information

Local information:

- $D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta)$: amount of info. at agent i to distinguish θ^* , θ
  - $D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta) = 0$: non-informative
  - $D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta) \neq 0$: informative



Global information:

- $\sum_i D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta)$: amount of info. globally available
  - $\sum_i D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta) = 0$: collectively non-informative
  - $\sum_i D_{KL}(\ell^i_{\theta*} \parallel \ell^i_\theta) \neq 0$: collectively informative

Question: Will collectively **non-informative** sufficient to learn θ^* ?

Network Identifiability Condition

- **every agent is cooperative**: "collectively informative" is sufficient, i.e., θ^* identifiable if

$$\sum_i D(\ell^i_{\theta^*} || \ell^i_{\theta}) \neq 0, \qquad \forall \theta \neq \theta^*$$

Network Identifiability Condition

- **every agent is cooperative**: “collectively informative" is sufficient, i.e., θ^* identifiable if

$$\sum_i D(\ell^i_{\theta*} || \ell^i_{\theta}) \neq 0, \qquad \forall\, \theta \neq \theta^*$$

- **Byzantine faults:** “collectively informative" is **NOT sufficient** !
  - Information propagation obstructed by Byzantine agents

Network Identifiability Condition

- **every agent is cooperative**: "collectively informative" is sufficient, i.e., θ^* identifiable if

$$\sum_i D(\ell^i_{\theta*} || \ell^i_{\theta}) \neq 0, \qquad \forall\, \theta \neq \theta^*$$

- **Byzantine faults:** "collectively informative" is **NOT sufficient** !
  - Information propagation obstructed by Byzantine agents

Stronger network identifiability is required !!!

Contributions

- First learning algorithm robust to Byzantine attacks:

Contributions

- First learning algorithm robust to Byzantine attacks:
  - each non-faulty agent learns the true state almost surely
  - beliefs on the wrong state decrease $O(\exp(-\tilde{C}t^2))$

Contributions

- First learning algorithm robust to Byzantine attacks:
  - each non-faulty agent learns the true state almost surely
  - beliefs on the wrong state decrease $O(\exp(-\tilde{C}t^2))$
- Identify sufficient condition on the global identifiability

Contributions

- First learning algorithm robust to Byzantine attacks:
  - each non-faulty agent learns the true state almost surely
  - beliefs on the wrong state decrease $O(\exp(-\tilde{C}t^2))$
- Identify sufficient condition on the global identifiability
- When $f = 0$ (failure-free): $O(\exp(-\tilde{C}t^2))$

Contributions

- First learning algorithm robust to Byzantine attacks:
  - each non-faulty agent learns the true state almost surely
  - beliefs on the wrong state decrease $O(\exp(-\tilde{C}t^2))$
- Identify sufficient condition on the global identifiability
- When $f = 0$ (failure-free): $O(\exp(-\tilde{C}t^2))$
- Low complexity variation
  - Complexity: $O(m^2 n \log n)$
  - Minimal global identifiability

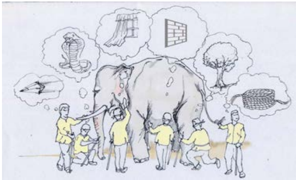
Related Work

Failure-free

- Bayesian learning: [Banerjee92, Gale03, Acemoglu11]
  - high complexity
- Non-Bayesian learning [Bala98, Acemoglu10, Golub10, Jadbabaie12]
  - Consensus-based models [Jadbabaie12] [Nedic, Olshevsky, Uribe, TAC'17]

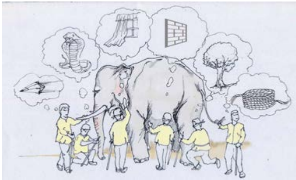
Belief Vectors

- $\mu_t^i = [\mu_t^i(\theta_1), \ldots, \mu_t^i(\theta_m)]$: approximate belief vector
- μ_0^i: initial belief



Belief Vectors

- $\mu_t^i = [\mu_t^i(\theta_1), \ldots, \mu_t^i(\theta_m)]$: approximate belief vector
- μ_0^i: initial belief
- $\boxed{\text{Goal: } \lim_t \mu_t^i(\theta^*) 1}$



θ_1	elephant
θ_2	spear
θ_3	snake
θ_4	curtain
θ_5	wall
θ_6	tree
θ_7	rope

μ_t^i(elephant)	0.3
μ_t^i(spear)	0.3
μ_t^i(snake)	0.1
μ_t^i(curtain)	0.1
μ_t^i(wall)	0.1
μ_t^i(tree)	0.05
μ_t^i(rope)	0.05

Our Algorithm

Network: Alice, Bob, Charlie and David

Our Algorithm

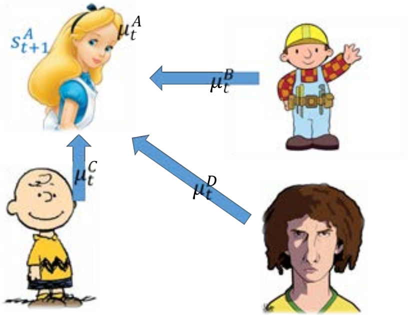
At the end of time t

Our Algorithm

During time $t + 1$: new observation

Our Algorithm

During time $t+1$: beliefs from neighbors



Our Algorithm

$\mu_t^A \quad \mu_t^B \quad \mu_t^C \quad \mu_t^D$

Update

$$\mu_{t+1}^A(\theta)$$

$$\text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\}$$

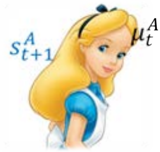
Our Algorithm

s_{t+1}^{A} μ_t^A μ_t^B μ_t^C μ_t^D

Update

$$\mu_{t+1}^{A}(\theta)$$
$$\text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^{\theta}(s_1^A, \cdots, s_{t+1}^A)$$

Our Algorithm



Update

$$\mu_{t+1}^{A}(\theta) \;\propto$$
$$\text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^{\theta}(s_1^A, \cdots, s_{t+1}^A)$$

Our Algorithm

Update

$$\mu_{t+1}^A(\theta) \ \propto$$
$$\text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^\theta(s_1^A, \cdots, s_{t+1}^A)$$

$\ell_A^\theta(s_1^A, \cdots, s_{t+1}^A)$:

- local history summary
- easy computation:
  $\ell_A^\theta(s_1^A, \cdots, s_{t+1}^A) = \ell_A^\theta(s_1^A, \cdots, s_t^A)\ell_A^\theta(s_{t+1}^A)$

Information Reconciliation

Update

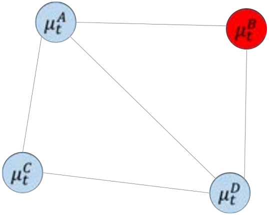
$$\mu^A_{t+1}(\theta) \;\propto\; \text{reconcile}\{\mu^A_t(\theta), \mu^B_t(\theta), \mu^C_t(\theta), \mu^D_t(\theta)\} \times \ell^\theta_A(s^A_1, \cdots, s^A_{t+1})$$

Information Reconciliation

Update

$$\mu_{t+1}^A(\theta) \propto \text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^\theta(s_1^A, \cdots, s_{t+1}^A)$$

1. malicious messages

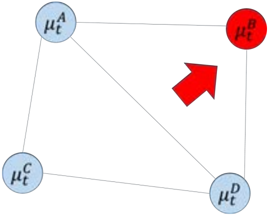


Information Reconciliation

Update

$$\mu_{t+1}^A(\theta) \propto \text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^\theta(s_1^A, \cdots, s_{t+1}^A)$$

1. malicious messages
2. beliefs can completely biased



Information Reconciliation

Update

$$\mu_{t+1}^A(\theta) \ \propto \ \text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^\theta(s_1^A, \cdots, s_{t+1}^A)$$

1. malicious messages
2. beliefs can completely biased
3. need to remove “outliers"

Information Reconciliation

Update

$$\mu_{t+1}^{A}(\theta) \;\propto\; \text{reconcile}\{\mu_t^{A}(\theta), \mu_t^{B}(\theta), \mu_t^{C}(\theta), \mu_t^{D}(\theta)\} \times \ell_A^{\theta}(s_1^A, \cdots, s_{t+1}^A)$$

1. malicious messages
2. beliefs can completely biased
3. need to remove “outliers"

Byzantine consensus: Trimming away “outliers" + averaging

[Mendes&Herlihy 2013, Vaidya&Garg 2013, Vaidya 2014]

Information Reconciliation

Update

$$\mu_{t+1}^{A}(\theta) \;\propto\; \text{reconcile}\{\mu_t^A(\theta), \mu_t^B(\theta), \mu_t^C(\theta), \mu_t^D(\theta)\} \times \ell_A^{\theta}(s_1^A, \cdots, s_{t+1}^A)$$

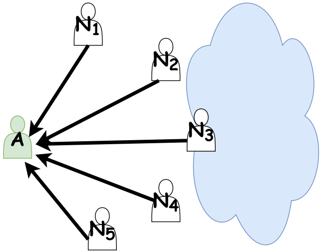
1. malicious messages
2. beliefs can completely biased
3. need to remove “outliers"

Byzantine consensus: Trimming away “outliers" + averaging

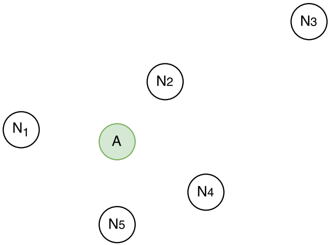
[Mendes&Herlihy 2013, Vaidya&Garg 2013, Vaidya 2014]

Byzantine consensus + local learning

Information propagation is RANDOM

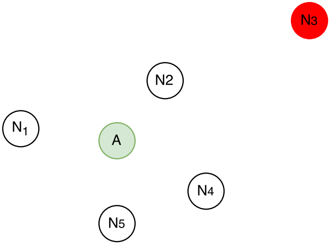


Information propagation is RANDOM



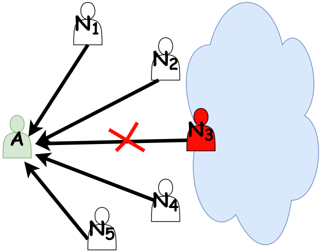
Belief vectors received by agent A

Information propagation is RANDOM



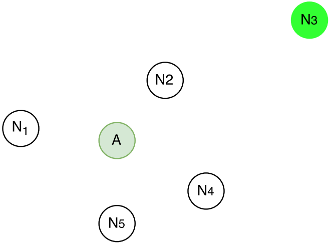
Belief vectors received by agent A

Information propagation is RANDOM



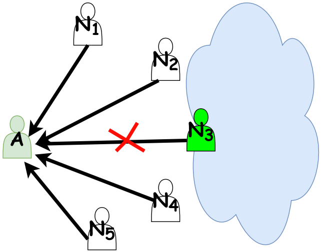
Information propagation forbidden due to Byzantine behaviors

Information propagation is RANDOM



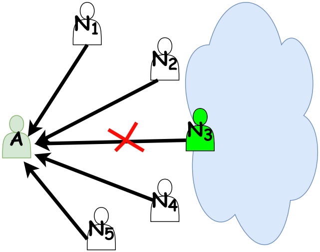
Belief vectors received by agent A

Information propagation is RANDOM



Information propagation forbidden due to randomness of local beliefs

Information propagation is RANDOM



Information propagation forbidden due to randomness of local beliefs

> Info. propagation inherits randomness from local observations

Existing analysis does not applicable

Convergence Results

Theorem

Under some network identifiability condition, for $\theta \neq \theta^*$,

$$\lim_{t} \mu_i^t(\theta^*) 1$$

Corollary (Convergence rate)

$$\mu_t^i(\theta) \leq \exp\left(-Ct^2\right) \quad a.s. \, (C > 0)$$

Sufficient Network Identifiability Condition

- Communication network not reflect the real info. flow
  - Information propagation interfered by Byzantine agents

Sufficient Network Identifiability Condition

- Communication network not reflect the real info. flow
  - Information propagation interfered by Byzantine agents
- Effective communication network
  - Information source agents $\mathcal{S}$: propagate info. out
  - Observations of agents to be collectively informative, i.e.,

$$\sum_{i \in \mathcal{S}} D_{KL}\left(\ell^i_{\theta^*} \parallel \ell^i_{\theta}\right) \neq 0 \;\; \forall \theta \neq \theta^* \tag{2}$$

Sufficient Network Identifiability Condition

- Communication network not reflect the real info. flow
  - Information propagation interfered by Byzantine agents
- Effective communication network (multiple)
  - Information source agents $\mathcal{S}$: propagate info. out
  - Observations of agents to be collectively informative, i.e.,

$$\sum_{i \in \mathcal{S}} D_{KL}\left(\ell^i_{\theta^*} \parallel \ell^i_{\theta}\right) \neq 0 \;\; \forall \theta \neq \theta^* \tag{2}$$

Sufficient Network Identifiability Condition

- Communication network not reflect the real info. flow
  - Information propagation interfered by Byzantine agents
- Effective communication network (multiple)
  - Information source agents $\mathcal{S}$: propagate info. out
  - Observations of agents to be collectively informative, i.e.,

$$\sum_{i \in \mathcal{S}} D_{KL}\left(\ell^i_{\theta^*} \parallel \ell^i_{\theta}\right) \neq 0 \;\; \forall \theta \neq \theta^* \tag{2}$$

Sufficient Network Identifiability Condition

For every effective communication network, (2) is satisfied

Comparison with Existing Failure-Free Algrorithm

Our algorithm

- Update rule:
  $\mu^i_{t+1}(\theta) \propto \text{averaging}\{\mu^j_t(\theta), j \in \mathcal{I}_i\} \times \ell^\theta_i(s^i_1, \ldots, s^i_{t+1})$
- Convergence rate: $\mu^i_t(\theta) \le \exp\left(-Ct^2\right)$

Existing algorithm [Jadbabaie 12, Shahrampour 15, Nedic 15]

- Update rule: $\mu^i_{t+1}(\theta) \propto \text{averaging}\{\mu^j_t(\theta), j \in \mathcal{I}_i\} \times \ell^\theta_i(s^i_{t+1})$
- Convergence rate: $\mu^i_t(\theta) \le \exp\left(-\tilde{C}t\right)$

Low Complexity Variant

- Computation complexity per iteration: High
- Network identifiability: Not minimal

Low Complexity Variant

- Computation complexity per iteration: High
- Network identifiability: Not minimal

Low Complexity Variant

m-ary hypo. testing $\Rightarrow$ $m(m-1)$ ordered binary hypo. testing

For each pair θ_1 and θ_2, update the likelihood ratio of θ_1 over θ_2

Low Complexity Variant

- Computation complexity per iteration: High
- Network identifiability: Not minimal

Low Complexity Variant

m-ary hypo. testing $\Rightarrow$ $m(m-1)$ ordered binary hypo. testing

For each pair θ_1 and θ_2, update the likelihood ratio of θ_1 over θ_2

- Complexity: $O(m^2 n \log n)$
- Minimal network identifiability

Finite-time Guarantees for Byzantine-Resilient Distributed State Estimation with Noisy Measurements

Problem Formulation

State estimation: A static state $\theta^* \in \mathbb{R}^d$ that each of the non-Byzantine agent is interested in learning.

Constraints: an agent can collect *partial* and *noisy* measurements only.

- (Linear observation model) For each agent, its local measurement $y_i(t)$ at time t is generated as

$$y_i(t) := H_i\theta^* + w_i(t),$$

where

(1) $H_i \in \mathbb{R}^{n_i \times d}$, where $n_i \ll d$, is the local observation matrix

(2) $w_i(t)$'s are the observation noises that are zero mean and bounded. The observation noises across agents are independent.

Applications: IoT, machine learning, wireless networks, sensor networks, and robotic networks

Problem Formulation

State estimation: A static state $\theta^* \in \mathbb{R}^d$ that each of the non-Byzantine agent is interested in learning.

Constraints: an agent can collect *partial* and *noisy* measurements only.

- (Linear observation model) For each agent, its local measurement $y_i(t)$ at time t is generated as

$$y_i(t) := H_i \theta^* + w_i(t),$$

where

(1) $H_i \in \mathbb{R}^{n_i \times d}$, where $n_i \ll d$, is the local observation matrix

(2) $w_i(t)$'s are the observation noises that are zero mean and bounded. The observation noises across agents are independent.

*The local observation of a Byzantine agent is well-defined.

Related Work

- **Adversary-resilient State Estimation**

  There is a rich line of work on the adversary-resilient state estimation problem wherein the existence of a fusion center is assumed. [Kosut-Jia-Thomas-Tong '11] [Kim and Poor '11] [Sou-Sandberg-Johansson '13] . . .

- **Adversary-resilient Distributed State Estimation**

  [Sundaram-Hadjicostis '11] [Chen-Kar-Moura '18 a, b,c,d,e] [Mitra-Sundaram '18] [Mitra-Ghawash-Sundaram-Abbas '21]. . .

Related Work

- **Adversary-resilient State Estimation**

  There is a rich line of work on the adversary-resilient state estimation problem wherein the existence of a fusion center is assumed. [Kosut-Jia-Thomas-Tong '11] [Kim and Poor '11] [Sou-Sandberg-Johansson '13] ...
- **Adversary-resilient Distributed State Estimation**

  [Sundaram-Hadjicostis '11] [Chen-Kar-Moura '18 a, b,c,d,e] [Mitra-Sundaram '18] [Mitra-Ghawash-Sundaram-Abbas '21]...

Our focus:

Noisy measurements, partially observable local matrix, and finite-time guarantees.

A Distributed Optimization Prospective: Asymptotic local function

For each agent $i \in \mathcal{V}$, define its *asymptotic* local function $f_i : \mathbb{R}^d \to \mathbb{R}$ as

$$f_i(x) := \frac{1}{2}\mathbb{E}\left[\|H_i x - y_i\|_2^2\right],$$

where the expectation of $f_i(x)$ is taken over the randomness of w_i.

1* f_i is well-defined for each agent regardless of whether it is a good agent or a Byzantine agent

2* Since the distribution of w_i is unknown to agent i, at any finite t, function f_i is not accessible to agent i.

A Distributed Optimization Prospective: Finite-time local function

The agent has access to the *finite-time* or *empirical* local function $f_{i,t}$ defined as

$$f_{i,t}(x) := \frac{1}{2t}\sum_{s=1}^{t} \|H_i x - y_i(s)\|_2^2 ,$$

whose gradient at x is

$$\begin{aligned}\nabla f_{i,t}(x) &= \frac{1}{t}\sum_{s=1}^{t} H_i^\top \left(H_i x - y_i(s)\right) \\ &= H_i^\top H_i(x-\theta^*) - H_i^\top \frac{1}{t}\sum_{s=1}^{t} w_i(s).\end{aligned}$$

A First Thought?

Question: Combine the local gradient descent with multi-dimensional Byzantine resilient consensus?

- The computation complexity of the relevant consensus component is prohibitively high
  - which typically relies on using Tverberg points
- assured convergence rate scales poorly in d

Proposed Algorithm

High-level idea:

Each good agent iteratively aggregates the received messages by, for each coordinate, discarding the largest b and the smallest b values, and averaging the remaining.

- *Local gradient descent:* Agent i first computes the noisy local gradient $\nabla f_{i,t}(x_i(t-1))$, and performs local gradient descent to obtain $z_i(t)$, i.e.,

$$z_i(t) = x_i(t-1) - \nabla f_{i,t}(x_i(t-1)).$$

Proposed Algorithm (continued)

- *Information exchange:* It exchanges $z_i(t)$ with other agents in its local neighborhood. Recall that $m_{ij}(t) \in \mathbb{R}^d$ is the message sent from agent i to agent j at time t. It relates to $z_i(t)$ as follows:

$$m_{ij}(t) = \begin{cases} z_i(t) & \text{if } i \in (\mathcal{V}/\mathcal{A}); \\ \star & \text{if } i \in \mathcal{A}, \end{cases}$$

  where $\star$ denotes an arbitrary value.
- *Robust aggregation:* For each coordinate $k = 1, \ldots, d$, the agent computes the trimmed mean to obtain $x_i(t)$.

Main results: Complete graphs

for ease of illustration: Applicable to computer networks and wireless networks with message forwarding

Lemma

For each iteration t, each good agent $i \in \mathcal{V}/\mathcal{A}$, and each k, there exist coefficients $\left(\beta_{ij}^{k}(t),\ j \in \mathcal{V}/\mathcal{A}\right)$ such that

- $x_i^k(t) = \sum_{j \in \mathcal{V}/\mathcal{A}} \beta_{ij}^k(t) \left\langle z_j(t), e_k \right\rangle$;
- $0 \leq \beta_{ij}^k(t) \leq \frac{1}{\phi - b}$ *for all* $j \in \mathcal{V}/\mathcal{A}$ *and* $\sum_{j \in \mathcal{V}/\mathcal{A}} \beta_{ij}^k(t) = 1$,

where $\phi = |\mathcal{V}/\mathcal{A}|$.

Main results: Complete graphs

for ease of illustration: Applicable to computer networks and wireless networks with message forwarding

Lemma

For each iteration t, each good agent $i \in \mathcal{V}/\mathcal{A}$, and each k, there exist coefficients $\left(\beta_{ij}^k(t),\ j \in \mathcal{V}/\mathcal{A}\right)$ such that

- $x_i^k(t) = \sum_{j \in \mathcal{V}/\mathcal{A}} \beta_{ij}^k(t) \left\langle z_j(t), e_k \right\rangle$;
- $0 \le \beta_{ij}^k(t) \le \frac{1}{\phi - b}$ *for all* $j \in \mathcal{V}/\mathcal{A}$ *and* $\sum_{j \in \mathcal{V}/\mathcal{A}} \beta_{ij}^k(t) = 1$,

where $\phi = |\mathcal{V}/\mathcal{A}|$.

Observations

- The update of x_i uses the information provided by the *good* agents only;
- each of the good agent has limited impact on x_i;

Remaining analysis is still non-trivial because

$\left(\beta_{ij}^k(t),\ j \in \mathcal{V}/\mathcal{A}\right) \neq \left(\beta_{ij}^{k'}(t),\ j \in \mathcal{V}/\mathcal{A}\right)$ for $k \neq k'$

Main results: Complete graphs

Assumption 1

For all $k = 1, \cdots, d$, we have that

$$\frac{1}{\phi - b} \sum_{j \in \mathcal{V}/\mathcal{A}} \left\| \left(\mathbf{I} - H_j^\top H_j \right) e_k \right\|_1 < 1.$$

- Note that it can well be the case that $\left\| \left(\mathbf{I} - H_j^\top H_j \right) e_k \right\|_1 \geq 1$ for some good agents.
- None of the agents are required to satisfy $\left\| \left(\mathbf{I} - H_j^\top H_j \right) e_k \right\|_1 < 1$ simultaneously for all $k = 1, \cdots, d$.

Main theorem

Let $\rho \triangleq \max_{k:1\le k\le d} \frac{\sum_{j\in\mathcal{V}/\mathcal{A}} \left\| \left(\mathbf{I} - H_j^\top H_j\right) e_k \right\|_1}{\phi - b}$, and
$C_0 \triangleq \max_{i\in\mathcal{V}/\mathcal{A}} \|H_i\|_2$.

Theorem

Suppose Assumption 1 holds and the graph is complete. Then

$$\max_{i\in\mathcal{V}/\mathcal{A}} \|x_i(t) - \theta^*\|_\infty \overset{a.s.}{\to} 0.$$

Moreover, with probability at least
$1 - \phi \exp\left(\frac{-\epsilon^2(1-\rho)^2 t}{8C^2}\right)$*, it holds that*

$$\max_{i\in\mathcal{V}/\mathcal{A}} \|x_i(t) - \theta^*\|_\infty \le \rho^t \max_{i\in\mathcal{V}/\mathcal{A}} \|x_i(0) - \theta^*\|_\infty$$
$$+ C_0 \left(\sum_{i\in\mathcal{V}/\mathcal{A}} \sqrt{\mathrm{trace}(\Sigma_j)} \right) \sum_{m=1}^{t-1} \frac{\rho^m}{\sqrt{t-m}} + \phi\epsilon.$$

Main results: Incomplete graphs

Theorem

Under the assumption that ensures Byzantine consensus with scalar inputs, if an assumption analogous to Assumption 1 holds, then any given $\delta \in (0, 1)$*, any* $\epsilon > 0$*, and*

$$t \geq \Omega\left(n^2/\epsilon^2\right)\left(\log \frac{1}{\delta} + \log n\right),$$

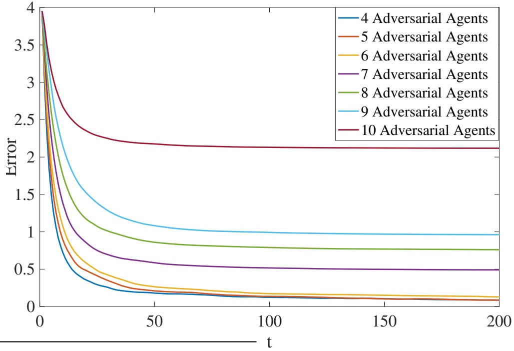
with probability at least $1 - \delta$*, it holds that*

$$\max_{i \in \mathcal{V}/\mathcal{A}} \|x_i(t) - \theta^*\|_\infty \leq \tilde{\rho}^t \max_{i \in \mathcal{V}/\mathcal{A}} \|x_i(0) - \theta^*\|_\infty + \tilde{C}_0 n \sum_{m=1}^{t-1} \frac{\tilde{\rho}^m}{\sqrt{t-m}} + \epsilon,$$

where $\tilde{\rho} \in (0, 1)$*.*

Numerical Examples: Energy Efficiency Data Set

- Regression dataset on UCI Machine Learning Repository[1]: In this dataset, the vector $\theta^* \in \mathbb{R}^8$, including eight features.
- We consider a network of $|\mathcal{V} \setminus \mathcal{A}| = 160$ agents. Each agent *i* observes only one feature corrupted by a Gaussian noise $\mathcal{N}(0, 0.25)$. Also, each agent *i* is connected to 40 agents $i-20, i-19, \ldots, i+19, i+20$.



[1] https://archive.ics.uci.edu/ml/datasets/Energy+efficiency

What we discussed

- Review: Faulty free
- Crash failure and Byzantine-resilience
- Impossibility results for Byzantine-resilience
- Algorithms for Byzantine-resilience
- Optimization problem with additional structures