

Decentralized Collaborative Learning with Probabilistic Data Protection

Tsuyoshi Idé (“Ide-san” 井手 剛)¹, Rudy Raymond²

¹ IBM Research, T. J. Watson Research Center (New York)

² IBM Research - Tokyo

Agenda

- Blockchain as value co-creation platform
- Decentralized multi-task learning: Problem setting
- Secure decentralized aggregation
- Network topology design
- Future research topics

1st gen Blockchain: Designed specifically for currency transfer



■ Blockchain 1.0: Bitcoin

- Designed specifically for currency transfer
- Verifying a transaction is trivial: just by checking account balances
- A unique consensus algorithm is used (“proof-of-work”)

■ Limitations

- Unable to handle general business transactions
- Proof-of-work lacks a deterministic guarantee



2nd gen Blockchain: General-purpose business transaction management platform



- Blockchain 2.0: Smart-Contract-enabled transaction management platform
 - Designed to be able to handle “general” business transactions
 - Traditional consensus algorithm (e.g. PBFT) is typically used
- Limitations
 - Validating smart contracts is not straightforward (c.f. money transfer)
 - No “knowledge discovery” elements: only perform predefined routines



ethereum

3rd gen Blockchain: Towards AI-integrated value co-creation platform



- Blockchain 3.0: Value co-creation platform
- “Value co-creation”: Share data, and collaboratively develop new insights that cannot be accessed when looking at your own data alone
- AI/machine learning provides a systematic means for value co-creation

Three requirements of value co-creation platform: Democracy, diversity, privacy

■ Democracy

- All participants are equal
- No dictator/central server that controls everything

■ Diversity

- All participants are not the same
- They wish to have insights customized to each

■ Privacy

- All participants are allowed to keep own data secret
- Collaborative learning is not communism

Agenda

- Blockchain as value co-creation platform
- Decentralized multi-task learning: Problem setting
- Secure decentralized aggregation
- Network topology design
- Future research topics

These three requirements are naturally translated into specific machine learning problems

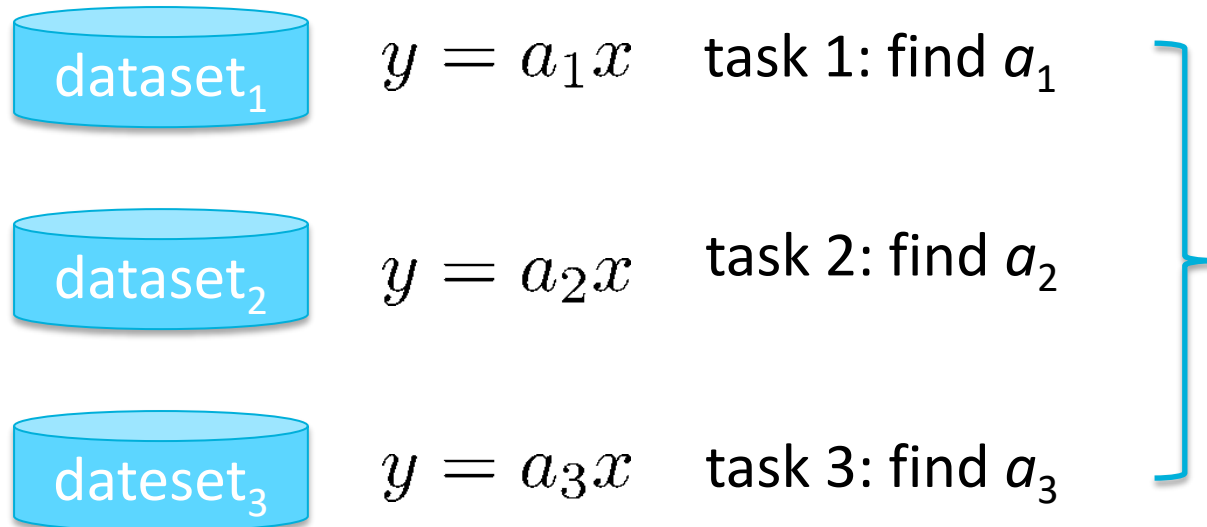
- Democracy → decentralized
 - All participants are equal
 - No dictator/central server that controls everything
- Diversity → multi-task
 - All participants are not the same
 - They wish to have insights customized to each
- Privacy
 - All participants are allowed to keep own data secret
 - Collaborative learning is not communism

**Blockchain 3.0
as
multi-task learning
with decentralization
and privacy
constraints**

(For ref.) Multi-task learning (MTL) is a framework that trains multiple models simultaneously under a relatedness constraint

- Multi-task = multiple models
- Key concept: task relatedness
 - Typically translated into a mathematical constraint on model parameters

- Linear regression example:



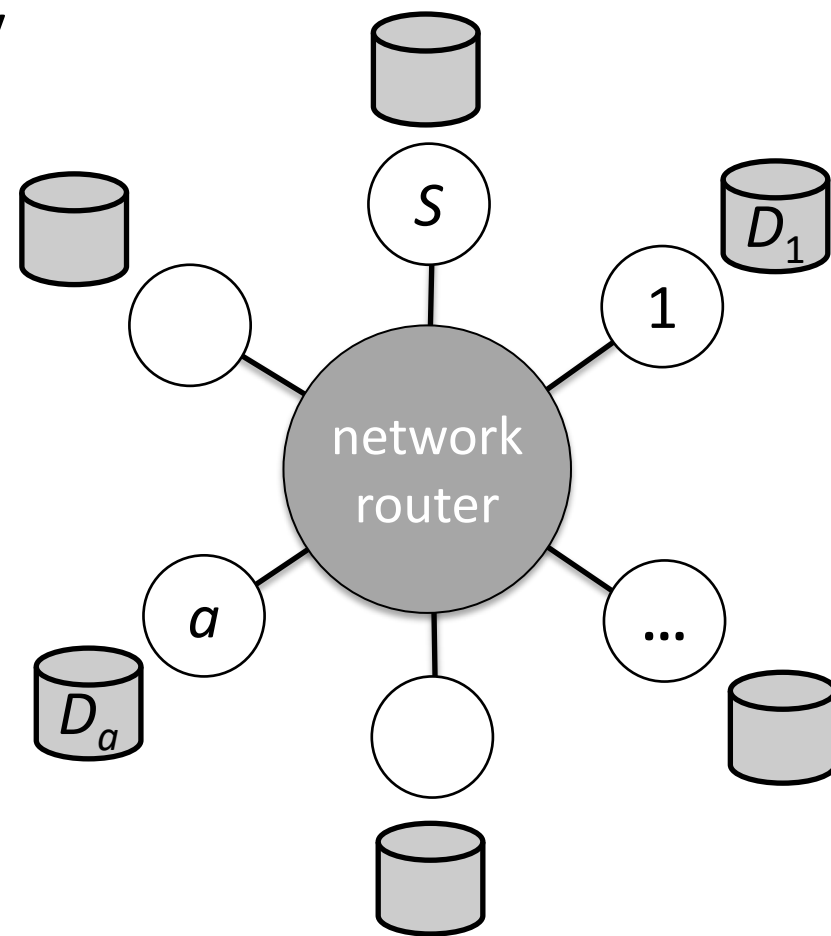
In MTL, we jointly learn $\{a_1, a_2, a_3\}$ under a constraint of relatedness.

Example:

$$\min_{a_1, a_2, a_3} \left\{ \text{squared loss} + \lambda \sum_{i,j} \underbrace{(a_i - a_j)^2}_{\text{relatedness constraint}} \right\}$$

We focus on multi-task density estimation as a concrete problem

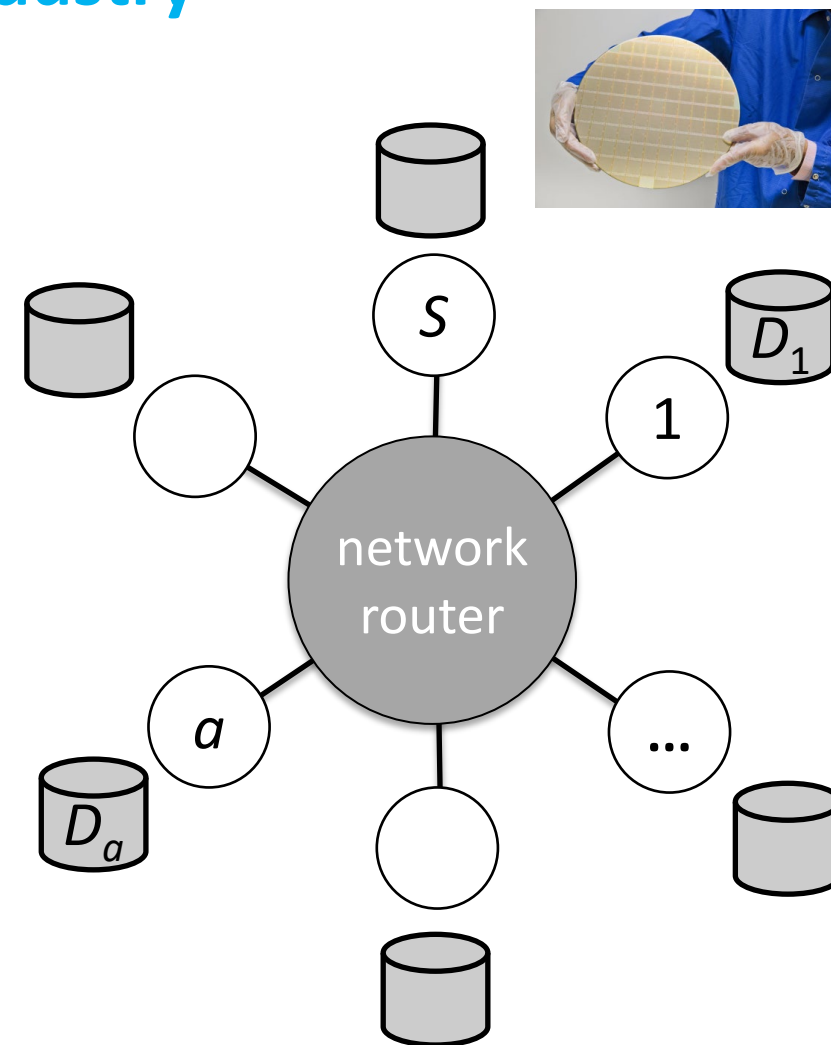
- Each participant ($a=1,.., S$) has a dataset D^a privately
 - $\mathcal{D}^a = \{\mathbf{x}^{a(1)}, \mathbf{x}^{a(2)}, \dots, \mathbf{x}^{a(N_a)}\}$
- The model in this case is the probability density function (pdf) of observed data \mathbf{x}
 - \mathbf{x} : real-valued multi-dimensional vector
- No central server. Only P2P communication is allowed according to a given network topology
- All the participants share the motivation of refining their model by leveraging other participants' knowledge



This setting is real:

Example from semiconductor manufacturing industry

- Density estimation $\hat{=}$ anomaly detection
 - low density region = unusual $\hat{=}$ anomalous
- Why multi-task?
 - The tools may be used in quite different manufacturing recipes. A one-size-fit-all solution won't apply.
- Why collaborative?
 - Anomalous samples are rare. The participants are eager to learn about potential anomalies that might occur in their own system.
- Why privacy/decentralized?
 - Information on failures is highly confidential. They don't want to disclose raw anomaly data.
 - They don't want send sensitive data to the server, either.

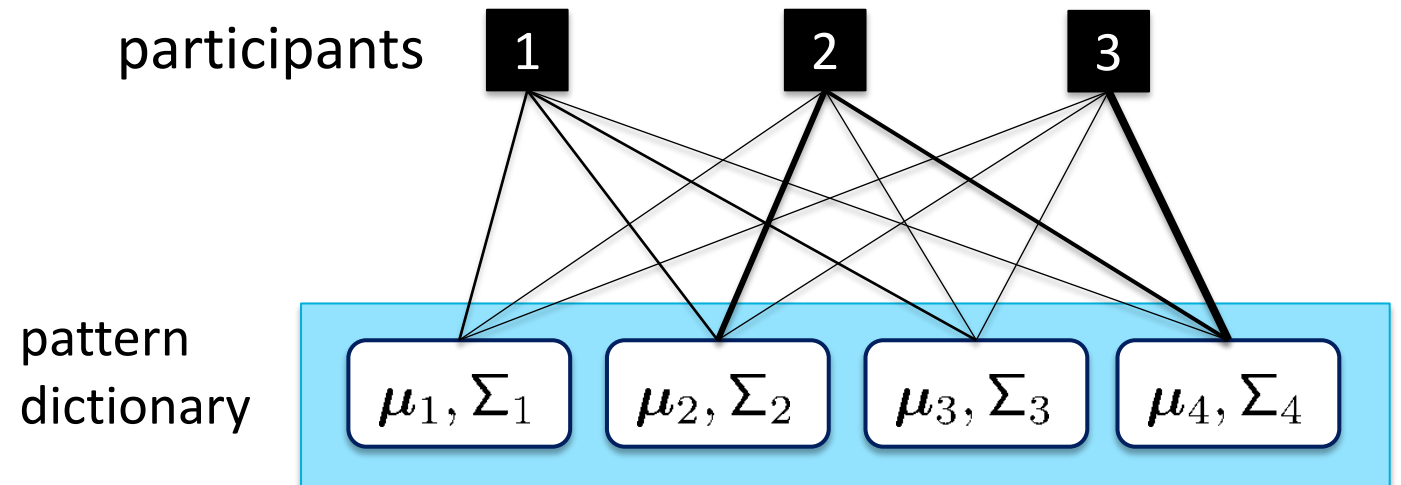


Formal problem description (details in the paper)

- Data: Participant a has a dataset D^a
 - $\mathcal{D}^a = \{\mathbf{x}^{a(1)}, \mathbf{x}^{a(2)}, \dots, \mathbf{x}^{a(N_a)}\}$
 - $\mathbf{x}^{a(n)}$ is the n -th sample of participant a (multivariate real-valued vector)
- Observation model: Probabilistic mixture model
 - Gaussian example (for participant a):
 - Prior distributions
$$p(\mathbf{x} \mid \Theta, \mathbf{u}^a) = \prod_{k=1}^K \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \Sigma_k)^{u_k^a}$$
 - ✓ Categorical-Dirichlet distribution for the one-hot cluster assignment variable \mathbf{u}^a
 - ✓ Laplace distribution for $(\Sigma_k)^{-1}$
- Inference approach: MAP (maximum a posteriori) with EM
 - The goal is to determine model parameters $\{(\boldsymbol{\mu}_1, \Sigma_1), \dots, (\boldsymbol{\mu}_K, \Sigma_K)\}$ (in the Gaussian example)
 - S participants collaboratively run an EM algorithm to maximize the likelihood function

How diversity is realized in the model while leveraging task relatedness: Illustration with $S=3$, $K=4$

- All the participants share the “pattern dictionary”, which stores $K=4$ sets of Gaussian parameters in this case
 - The participants collaboratively learn the dictionary so the total likelihood of the network is maximized
- Each of the participants gets a customized set of mixture weights, which represents diversity



Subproblems to achieve decentralized and privacy-preserving training

- The original maximum likelihood algorithm does not consider either decentralized or privacy-preserving aspects.
- We need a solution to:

**Decentralized
aggregation**

**Privacy breach
analysis**

**Network topology
design**

Agenda

- Blockchain as value co-creation platform
- Decentralized multi-task learning: Problem setting
- Secure decentralized aggregation
- Network topology design
- Future research topics

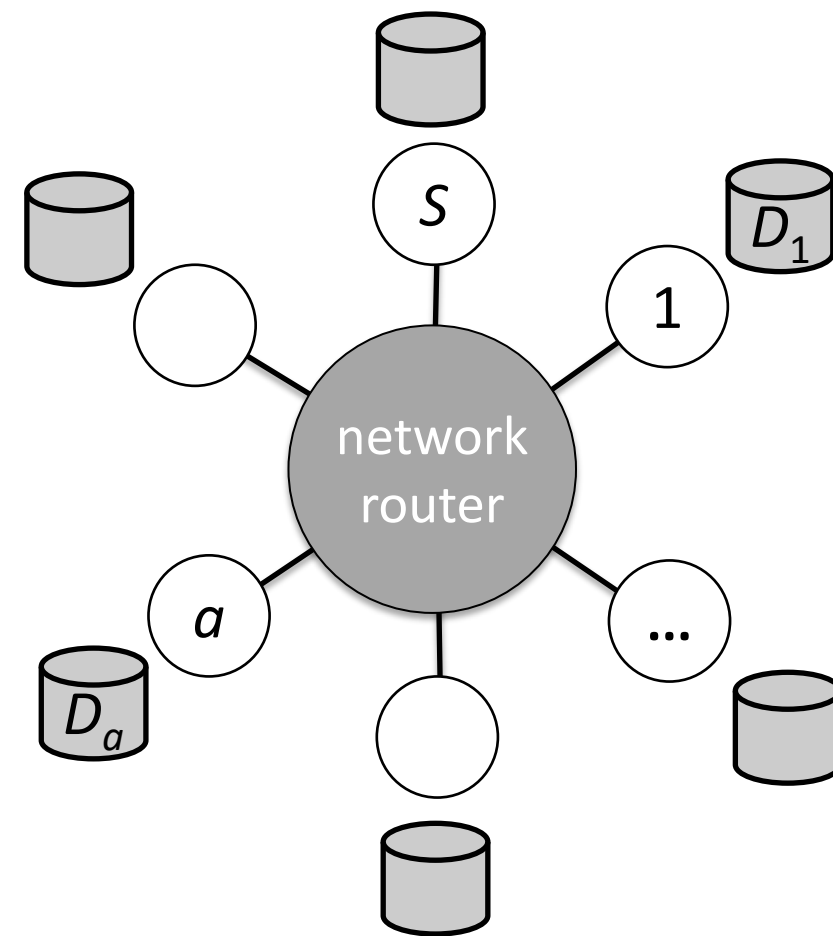
Secure aggregation problem

■ Problem: Compute summation

- $\bar{c} = c_1 + c_2 + \dots + c_S$
- c_a ($a=1, \dots, S$): A statistic (or a datum) computed locally by participant a

■ Easy? Not really, when only P2P communications are allowed

- Broadcasting your data to all?
 - ✓ No! Total privacy breach
- Select a leader to let her compute?
 - ✓ No! What if she is a bad guy?



Existing privacy-preservation approaches have issues in decentralized setting

- Encryption-based
 - Decentralization is nontrivial
 - Can be serious computational bottleneck
 - ✓ Great for one-time business transactions
 - ✓ Not designed for iterative machine learning algorithms
- "Noise-based" (differential privacy)
 - Typically needs central authority
 - Noise variance blows up in the multi-party setting as a result of aggregation
 - Learning models can be suboptimal due to noise

Our solution to secure aggregation problem

**Dynamical
consensus**

+

Secret sharing

- Repeat P2P communication so a certain Markov transition is performed
- Stationary state of the Markov chain converges to the aggregated value (magical!)

- Random chunking with probabilistic privacy guarantee
- Securer (but slow) alternative:
 - Shamir's secret sharing combined with dynamical consensus

Dynamical consensus algorithm: Leveraging Markovian dynamics for aggregation

- Algorithm: Each participant repeat an update until convergence

- $$c_a \leftarrow c_a + \epsilon \sum_{j=1}^S A_{a,j} (c_j - c_a)$$

Communicate only with connected peers

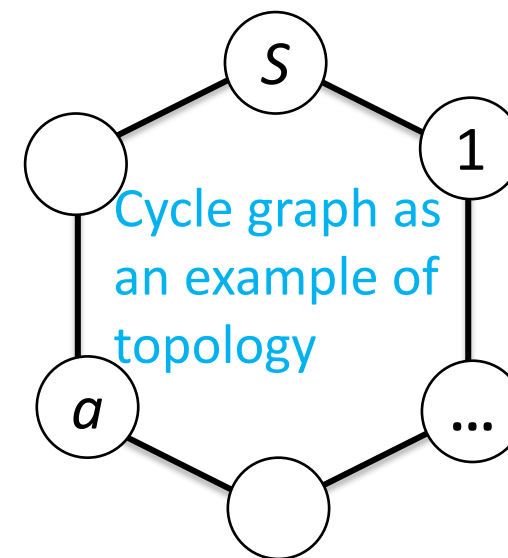
- A**: Network topology (= incidence matrix of the graph)
 - ϵ : A small positive constant

- Upon convergence, each participant ends up having

- $$\bar{c} = \sum_{a=1}^S c_a = \underline{\mathbf{1}}^\top \mathbf{c}$$

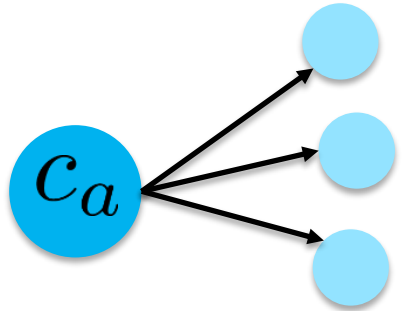
S -dimensional vector of ones

- Why? Because the update is the same as multiplying a matrix, whose leading eigenvector is the **1** vector. (\rightarrow see the paper)



Random chunking algorithm: Applying aggregation to each random split

- Each participant randomly splits their datum into N_c chunks

- $$\bar{c} = \sum_{a=1}^S c_a$$


$$c_a = c_a^{[1]} + c_a^{[2]} + c_a^{[3]}$$

- Do dynamic consensus N_c times and sum

- $$\bar{c} = \bar{c}^{[1]} + \bar{c}^{[2]} + \bar{c}^{[3]}$$

- Need to randomize node IDs every time upon starting aggregation
 - This is for a node not to receive all the chunks
 - Security guarantee becomes thus probabilistic

Random chunking algorithm trades off cryptographic security guarantee for computational efficiency

- Shamir's secret sharing (SSS) allows performing aggregation without revealing any raw data
- In random chunking, privacy guarantee is probabilistic. But it is a few orders of magnitude faster than SSS

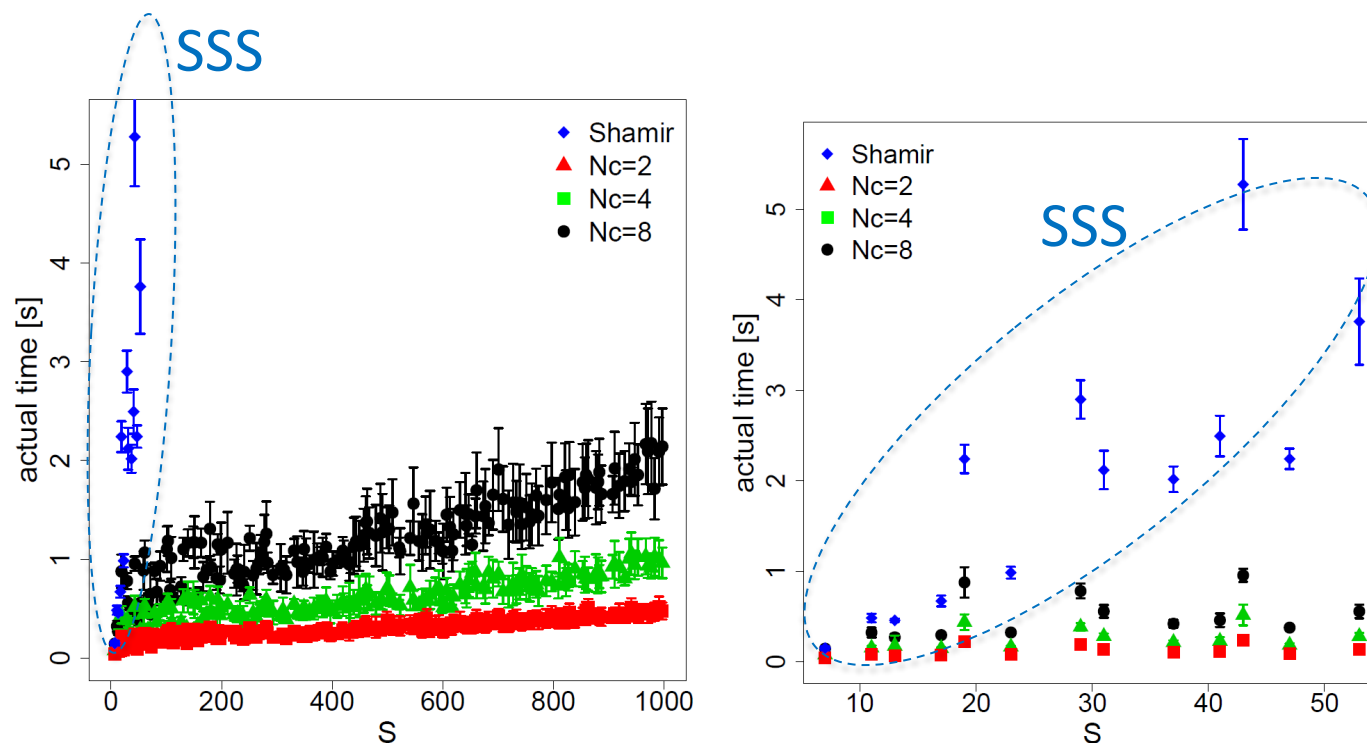


Fig. 6. Actual computation time for aggregation on the 3-regular expander graph. The right panel covers the range of $7 \leq S \leq 53$.

Probabilistic privacy guarantees of random chunking algorithm (proof → paper)

scenario	breach probability bound per node	parameters
independent	$\leq (S - 1) \left(\frac{d_a}{S - 1} \right)^{N_C}$	d_a : Node degree of the a -th node N_C : The number of splits
collusion	$\leq \exp \left\{ -N_C \left(1 - \frac{d_a}{S - N_L} \right)^{N_L} \right\}$	N_L : The number of colluded nodes
eavesdropping	$\leq \exp \left\{ -N_C \left(1 - \frac{N_E}{E - d_a + 1} \right)^{d_a} \right\}$	N_E : The number of tapped edges E : The total number of edges of the graph

Agenda

- Blockchain as value co-creation platform
- Decentralized multi-task learning: Problem setting
- Secure decentralized aggregation
- Network topology design
- Future research topics

Spectral stricture of \mathbf{W}_ϵ governs convergence speed in dynamical consensus

- The dynamical consensus algorithm can be viewed as repeated multiplication of a matrix $\mathbf{W}_\epsilon \equiv \mathbf{I} - \epsilon(\mathbf{D} - \mathbf{A})$
 - \mathbf{A} : adjacency matrix; \mathbf{D} : degree matrix
 - $\mathbf{D} - \mathbf{A}$ is known as the graph Laplacian
- The spectral structure of \mathbf{W}_ϵ governs convergence speed
 - Critical quantity is the “spectral gap”: $\lambda_1 - \lambda_2$
 - ✓ The difference between the 1st and the 2nd largest eigenvalues of \mathbf{W}_ϵ
- Question: How do I choose the topology so the spectral gap is as large as possible while keeping the probability of privacy breach low?

dynamical consensus update

$$c_a \leftarrow c_a + \epsilon \sum_{j=1}^S A_{a,j} (c_j - c_a)$$

Deep mathematical result in graph theory helps find a good compromise between privacy and convergence speed

- The topology should be
 - as sparse as possible for privacy protection
 - as dense as possible for faster convergence
- A class of graphs called the *expander graph* is an ideal compromise
 - Known as a sparse approximation of the complete graph
- Remarkable property of the expander graph
 - By Cheeger's inequality, we have

$$\Delta_\lambda \triangleq \lambda_1 - \lambda_2 \geq \epsilon \frac{\alpha^2}{2d} \quad (d\text{-regular expander graphs})$$

from which we can evaluate the number of iterations as

$$t \sim O\left(\frac{\ln(\sqrt{S}/\delta)}{|\ln(1 - \Delta_\lambda)|}\right) \quad \text{logarithmic convergence w.r.t. \# participants}$$

- α : lower bound of a quantity called the expansion coefficient
- δ : relative error allowed

(For ref.) Expansion coefficients and Cheeger's inequality

- Expander graph
 - A graph whose expanding constant is lower-bounded

- Expanding constant

- A measure of well-connectedness of a graph

$$\phi(G) \triangleq \inf_{\mathcal{V}_1} \frac{|\partial \mathcal{V}_1|}{\min\{|\mathcal{V}_1|, |G| - |\mathcal{V}_1|\}}$$

$\mathcal{V}_1 \in G$: arbitrary subgraph, $|\partial \mathcal{V}_1|$: #edges outgoing from \mathcal{V}_1

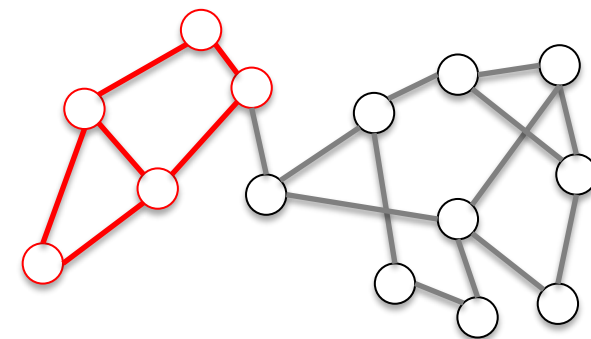
- Expander constant is known to be associated with graph spectrum

- Cheeger's inequality for d -regular expander graphs

$$\frac{\lambda_2}{2} \leq \frac{\phi(G)}{d} \leq \sqrt{2\lambda_2} \quad \lambda_2: \text{2nd eigenvalue of } D - A$$

- The lower-boundedness means a large spectral gap and thus fast convergence in the consensus algorithm

Example (low expansion const)



- $\mathcal{V}_1 = \text{red}$
- $|\mathcal{V}_1| = 5, \quad S \triangleq |G| = 14$
- $|\partial \mathcal{V}_1| = 1$

“Cycle with inverse chord” is one of few known constructions for expander graph

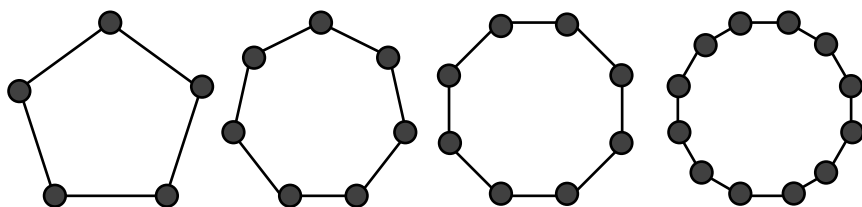
■ Cycle graph

○ Pros

- ✓ Sparse. Minimum number of neighbors. Good for privacy
- ✓ Symmetric. Regular. Good for democracy.
- ✓ Analytic expression of eigenvalues allows detailed convergence analysis

○ Cons

- ✓ Slow convergence.



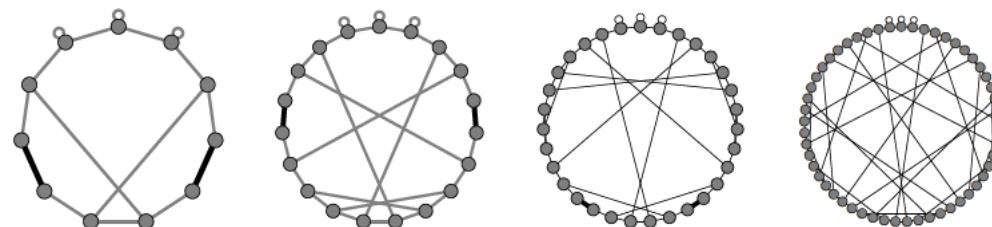
■ Cycle with inverse chord

○ Pros

- ✓ Reasonably sparse
- ✓ Fast convergence ($\sim \log S$)

○ Cons

- ✓ Not regular if S is non-prime
- ✓ Eigenvalue is non-smooth with S



Drastic speedup by expander graph

- Number of iteration is $\sim \log S$ when using the expander graph
 - S : The number of nodes (or network participants)
- Proposed method is several orders of magnitude faster than the existing fully-decentralized consensus algorithm using homomorphic encryption [Ruan+ 17;19]

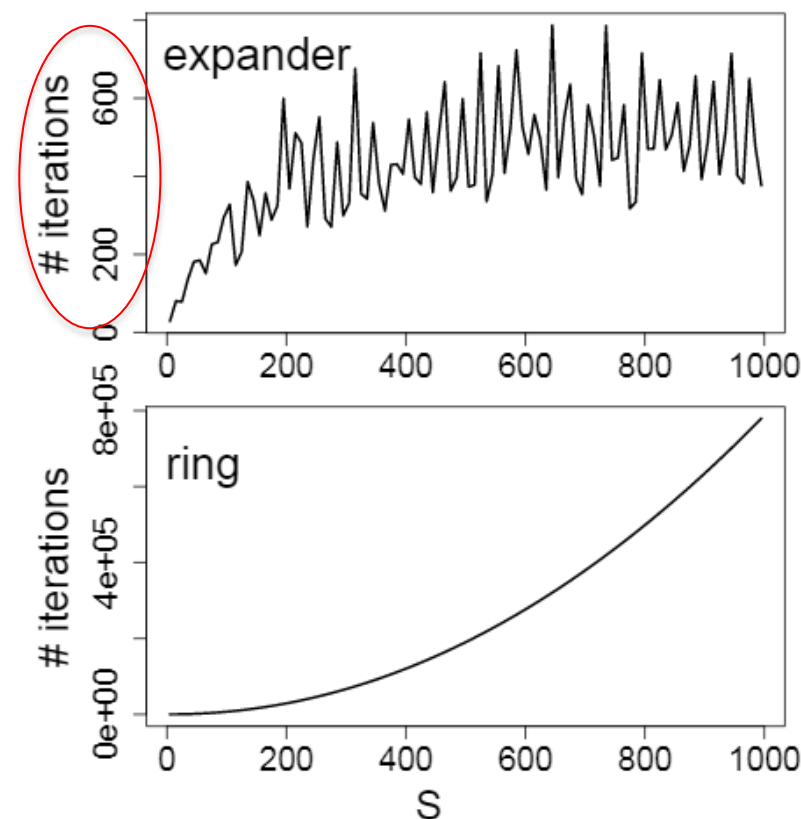


Fig. 5. Comparison of the number of iterations t for $\delta = 10^{-3}$.

Agenda

- Blockchain as value co-creation platform
- Decentralized multi-task learning: Problem setting
- Secure decentralized aggregation
- Privacy breach analysis
- Network topology design
- Future research topics

Future research topics

- Learning under network failures
 - The current model assumes perfect synchronization. Evaluating robustness under network failure and extending the algorithm to handle asynchronous communication is important.
- Meta-agreement issues
 - In addition to computed numerical statistics, there are several things that require participants' consensus
 - ✓ Choice of the algorithm, dimensionality, topology, etc.
- External data privacy
 - We focused on privacy guarantees among network participants. Evaluation of privacy leakage when, e.g., externally selling the learned model is an open question.
- Randomness in graph spectra
 - The expander graph provides an excellent convergence rate in dynamical consensus, but it introduces some unpredictability in the graph spectra.
- Security analysis
 - The random chunking algorithm combined with the dynamic consensus algorithm appears to have more flexibility than traditional cryptographic methods. We need to study further the pros and cons of those methods.
- Use-cases
 - Finally, we need to develop practical use-cases where the decentralized architecture is truly useful. The lightweight probabilistic privacy guarantee seems suitable in IoT applications, but more study is needed.

Thank you!

Appendix

- MAP framework detail

We employ a mixture of exponential family for multi-task density estimation

- Each agent holds its own data

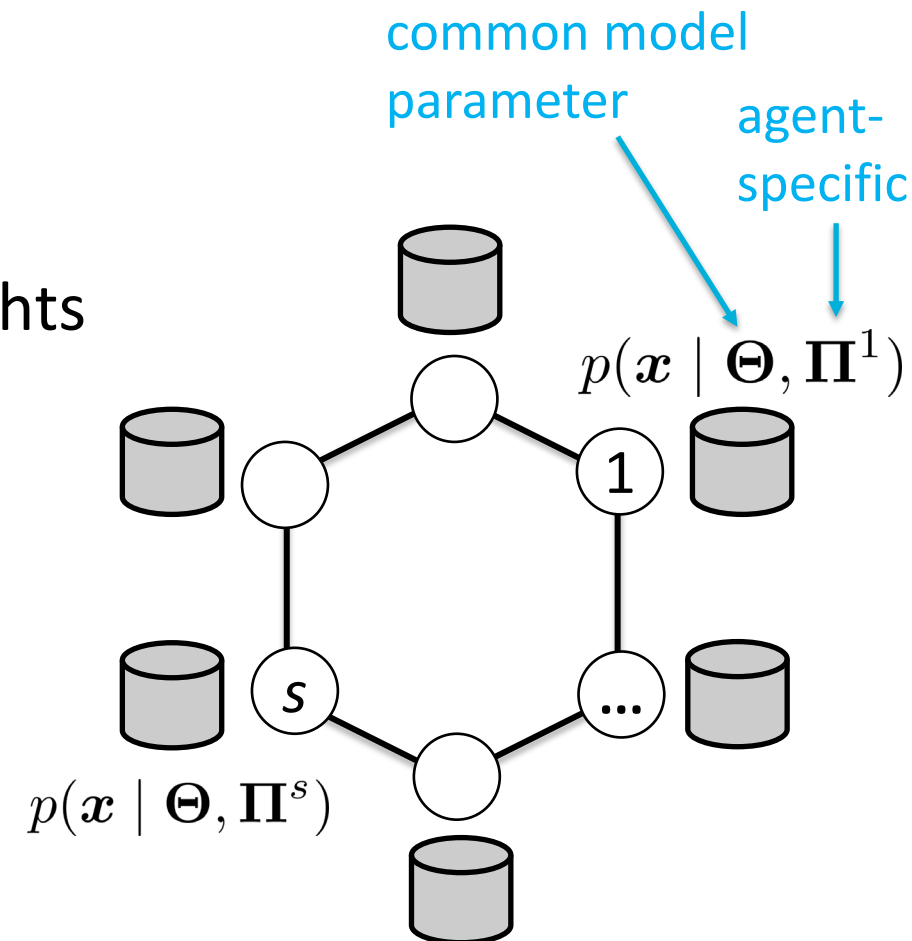
$$\mathcal{D}^s = \{\mathbf{x}^{s(n)} \mid n = 1, \dots, N^s; \mathbf{x}^{s(n)} \in \mathbb{R}^M\}$$

- Employ a mixture model with agent-specific weights

- $$p^s(\mathbf{x} \mid \Theta, \Pi^s) = \sum_{k=1}^K \pi_k^s f(\mathbf{x} \mid \theta_k)$$
- The mixture coefficients $\{\pi^1, \dots, \pi^S\}$ is agent-specific
- $\{\theta_1, \dots, \theta_K\}$ are shared by all the agents

- For f , employ exponential family

$$f(\mathbf{x} \mid \theta_k) = G(\theta_k) H(\mathbf{x}) \exp \{ \eta(\theta_k)^\top \mathbf{T}(\mathbf{x}) \}$$



Maximizing Jensen bound of log likelihood

- Observation model (for the s -th agent)

- $p(\mathbf{x} \mid \Theta, \mathbf{z}^s) = \prod_{k=1}^K f(\mathbf{x} \mid \theta_k)^{z_k^s}$ with $f(\mathbf{x} \mid \theta_k) = G(\theta_k)H(\mathbf{x}) \exp \{ \boldsymbol{\eta}(\theta_k)^\top \mathbf{T}(\mathbf{x}) \}$

- Prior distributions

- $p(\mathbf{z}^s \mid \boldsymbol{\pi}^s) = \text{Cat}(\mathbf{z}^s \mid \boldsymbol{\pi}^s) \equiv \prod_{k=1}^K (\pi_k^s)^{z_k^s}$, and $p(\Theta, \Pi) = \prod_{k=1}^K p(\theta_k) \prod_{s=1}^S p(\boldsymbol{\pi}^s)$

- Inference: maximize Jensen bound of log likelihood

- $\ln \sum_{\mathbf{Z}} p(\Pi, \Theta) \prod_{n,s} p(\mathbf{x}^{s(n)} \mid \Theta, \mathbf{z}^{s(n)}) p(\mathbf{z}^{s(n)} \mid \boldsymbol{\pi}^s)$
 - $\geq \text{c.} + \ln p(\Pi, \Theta) + \sum_{\mathbf{Z}} \textcolor{red}{Q}(\mathbf{Z}) \sum_{s,n} \ln [p(\mathbf{x}^{s(n)} \mid \Theta, \mathbf{z}^{s(n)}) p(\mathbf{z}^{s(n)} \mid \boldsymbol{\pi}^s)]$

- Maximizer for Q : $Q(\mathbf{Z}) = \prod_{n,s} \text{Cat}(\mathbf{z}^{s(n)} \mid \mathbf{r}^{s(n)}), \quad r_k^{s(n)} = \frac{\pi_k^s f(\mathbf{x}^{s(n)} \mid \theta_k)}{\sum_{m=1}^K \pi_m^s f(\mathbf{x}^{s(n)} \mid \theta_m)}$

Almost the same as
the standard EM
procedure of mixture
models

Exponential family naturally leads to Global-Local Separation in maximum likelihood

- The parameters $\{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}$ are point-estimated by maximizing the Jensen bound
- The solution has an interesting property: **Separation of local and global calculations**

- Local (in each agent): $\mathbf{T}_k^s \equiv \sum_{n=1}^{N^s} r_k^{s(n)} \mathbf{T}(\mathbf{x}^{s(n)}), \quad N_k^s \equiv \sum_{n=1}^{N^s} r_k^{s(n)}$

$p(\boldsymbol{\theta}_k)$ is the prior
of (common)
model parameter

- Global
 - ✓ Aggregation over the agents $N_k = \sum_{s=1}^S N_k^s, \quad \mathbf{T}_k = \sum_{s=1}^S \mathbf{T}_k^s$

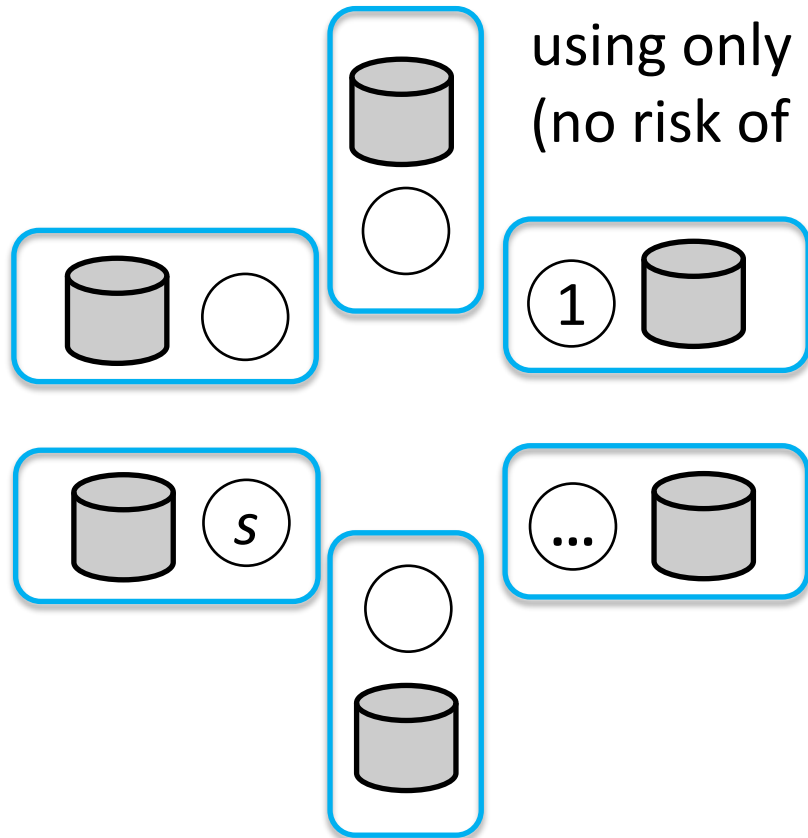
- ✓ Optimization $\boldsymbol{\theta}_k = \arg \max_{\boldsymbol{\theta}_k} \{ \ln p(\boldsymbol{\theta}_k) + N_k \ln G(\boldsymbol{\theta}_k) + \mathbf{T}_k^\top \boldsymbol{\eta}(\boldsymbol{\theta}_k) \}$

- Data privacy is concerned only with aggregation

The exponential family naturally leads to Global-Local Separation in maximum likelihood

Local updates:

compute statistics locally
using only my own data
(no risk of privacy breach)



Iterates until
convergence

Global consensus:

- Compute aggregation
- Perform optimization to store a unique result

