IJCAI-19
6042

Efficient Protocol for Collaborative Dictionary Learning in Decentralized Networks

T. Idé (井手 剛) [1], **Raymond Rudy[2]**, and **Dzung T. Phan[1]**

[1] IBM Research, T.J. Watson Research Center

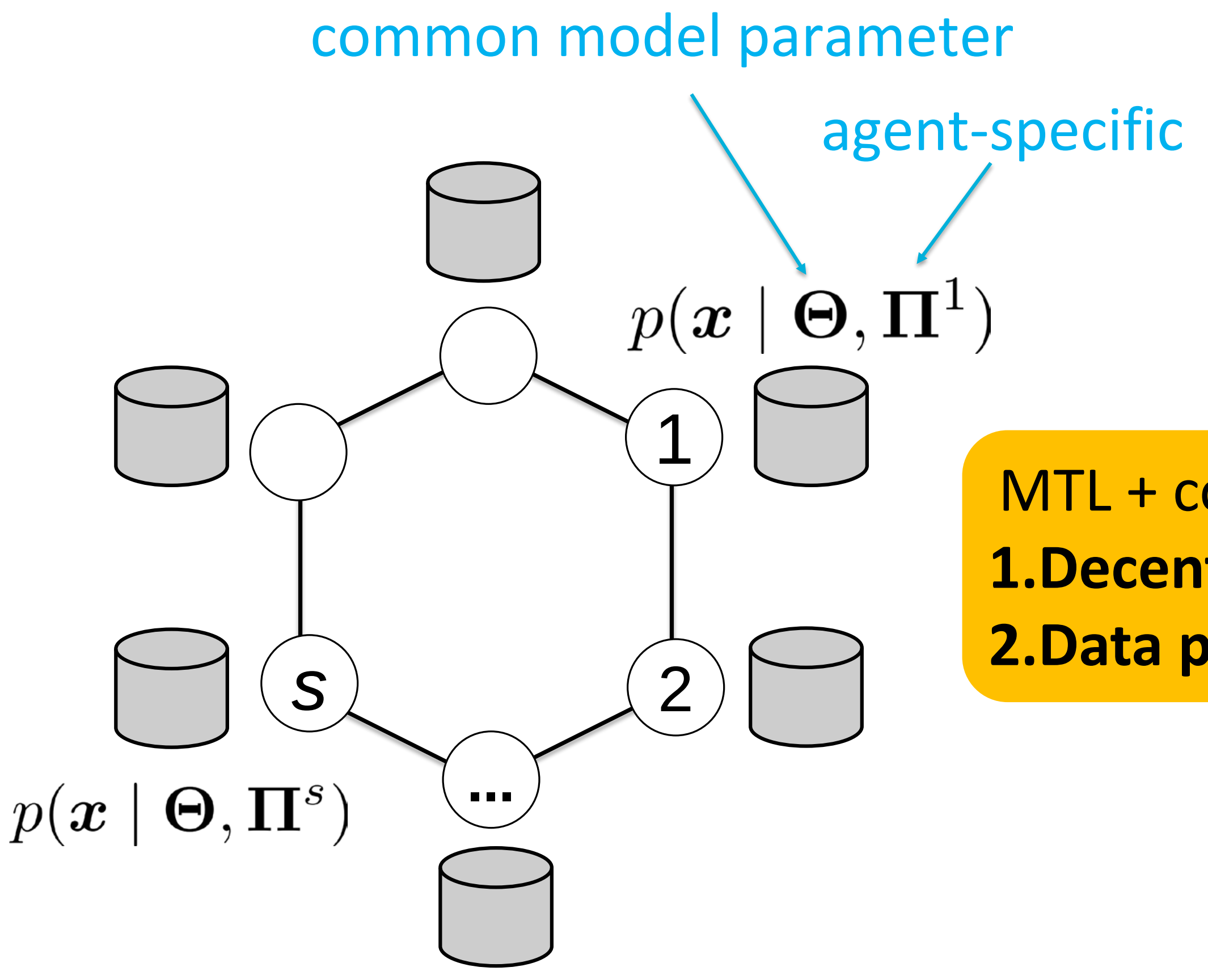
[2] IBM Research - Tokyo

The problem was inspired by

IBM Blockchain

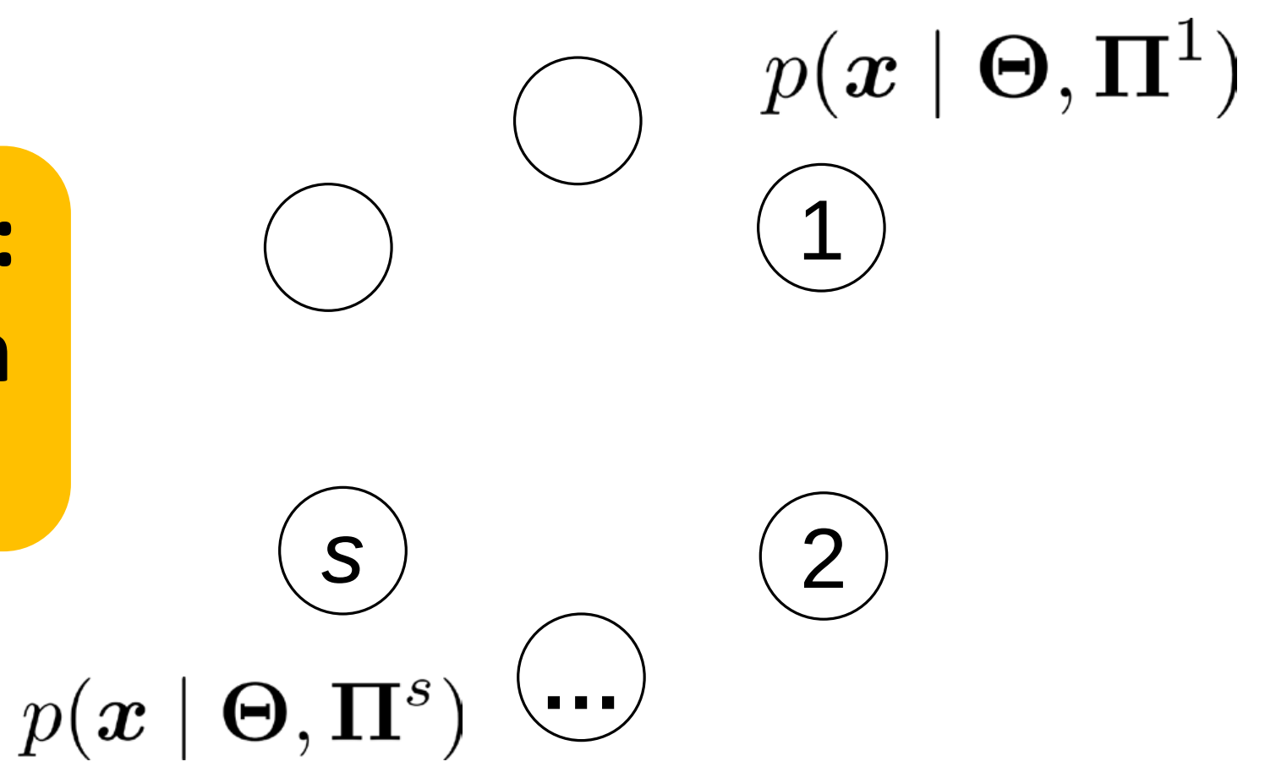
Problem setting

Decentralized multi-task density estimation with data privacy



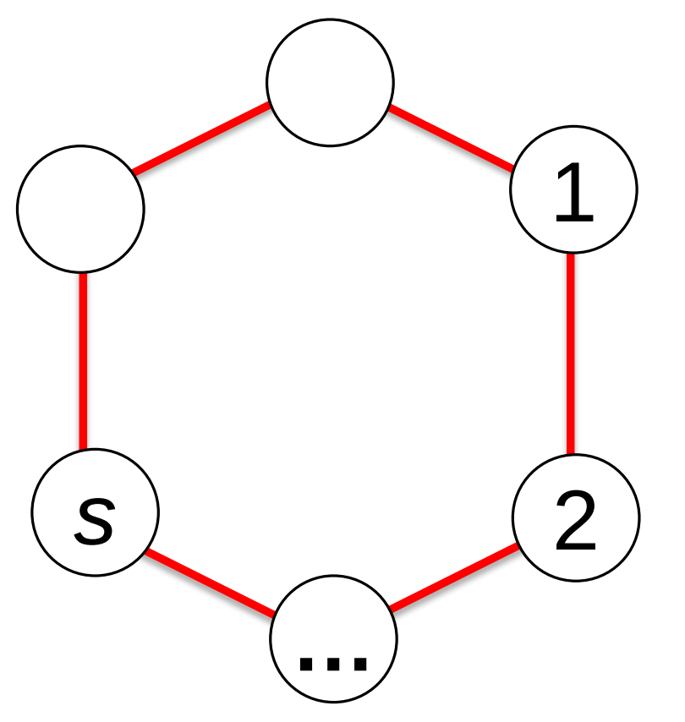
MTL + constraints:
1.Decentralization
2.Data privacy

Each agent wishes to learn its own probability density

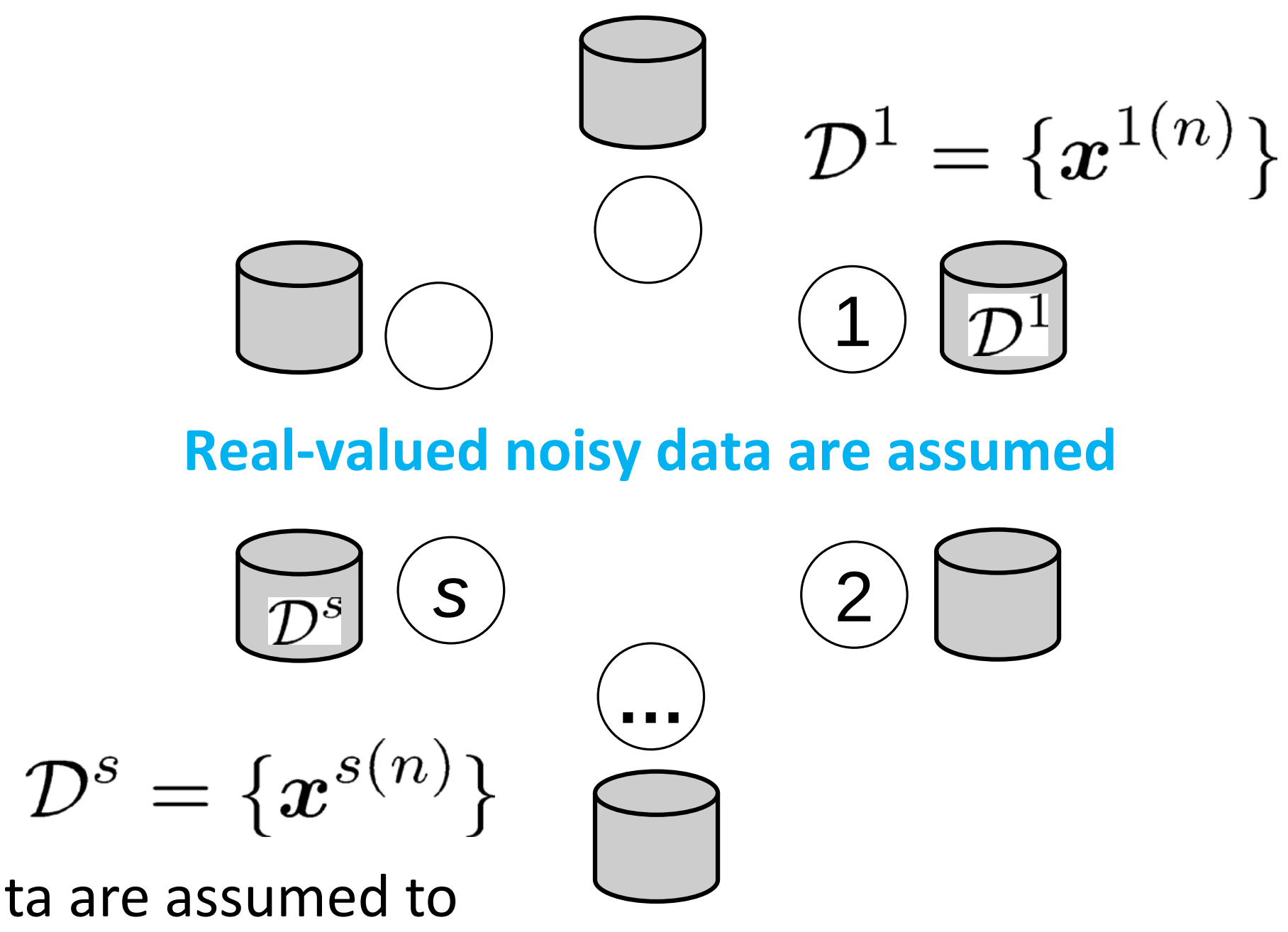


Closed, membership-based, and decentralized network

No central server!



Multiple "semi-honest" agents privately keep own data



Real-valued noisy data are assumed

All the data are assumed to have the same dimensionality

Prior work

Multi-task learning	Actively studied area but mostly for supervised learning Not many of them are fully probabilistic Little is known about how to decentralize it
Decentralized computation	Byzantine protocols assume categorical values Multi-agent consensus methods are not in the context of multi-task learning
Data privacy (under distributed environment)	Differential privacy is problematic in distributed environment Secure multi-party computation typically needs a central server Homomorphic encryption is slow

Multi-task density estimation model

Employ a mixture model with agent-specific weights

$$p^s(\boldsymbol{x} \mid \boldsymbol{\Theta}, \boldsymbol{\Pi}^s) = \sum_{k=1}^{K} \underline{\pi_k^s} f(\boldsymbol{x} \mid \boldsymbol{\theta}_k)$$

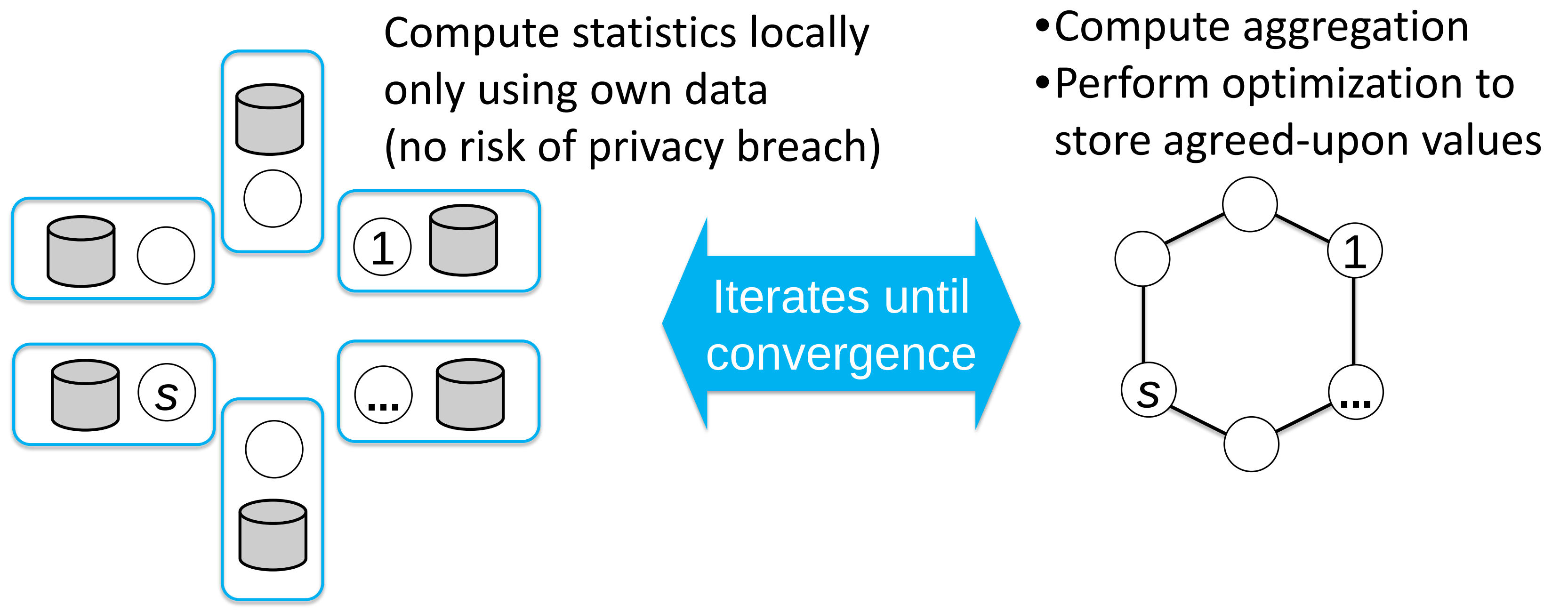
Shared by all the agents

Mixture weights are agent-specific

The density is assumed to be in the exponential family

$$f(\boldsymbol{x} \mid \boldsymbol{\theta}_k) = G(\boldsymbol{\theta}_k) H(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}(\boldsymbol{\theta}_k)^\top \boldsymbol{T}(\boldsymbol{x})\right\}$$

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



Local updates:
Compute statistics locally only using own data (no risk of privacy breach)

Global consensus:
- Compute aggregation
- Perform optimization to store agreed-upon values

Decentralized aggregation with data privacy

Decentralized aggregation
= Finding stationary state of Markovian transition process

Given the incidence matrix **A**, an update equation

$$c^s \leftarrow c^s + \epsilon \sum_{j=1}^{S} \mathsf{A}_{s,j}(c^j - c^s) \quad \text{or} \quad \boldsymbol{c} \leftarrow \mathbf{W}_\epsilon \boldsymbol{c}$$

converges to

$$\bar{c} = \sum_{s=1}^{S} c^s = \underline{\mathbf{1}}^\top \boldsymbol{c}$$

S-dimensional vector of ones

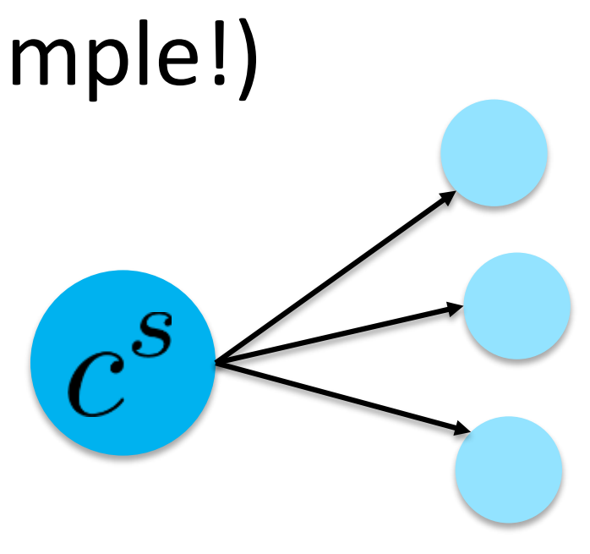
$$\mathbf{W}_\epsilon \equiv \mathbf{I} - \epsilon(\underline{\mathbf{D} - \mathbf{A}})$$

This is the graph Laplacian (minimum eigenvalue is 0)

"Chunking" method to prevent privacy breach

Randomly split each datum into N_c chunks and run aggregation algorithm N_c times (Simple!)

$$\bar{c} = \sum_{s=1}^{S} c^s$$



$$c^s = c^{s[1]} + c^{s[3]} + c^{s[3]}$$

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

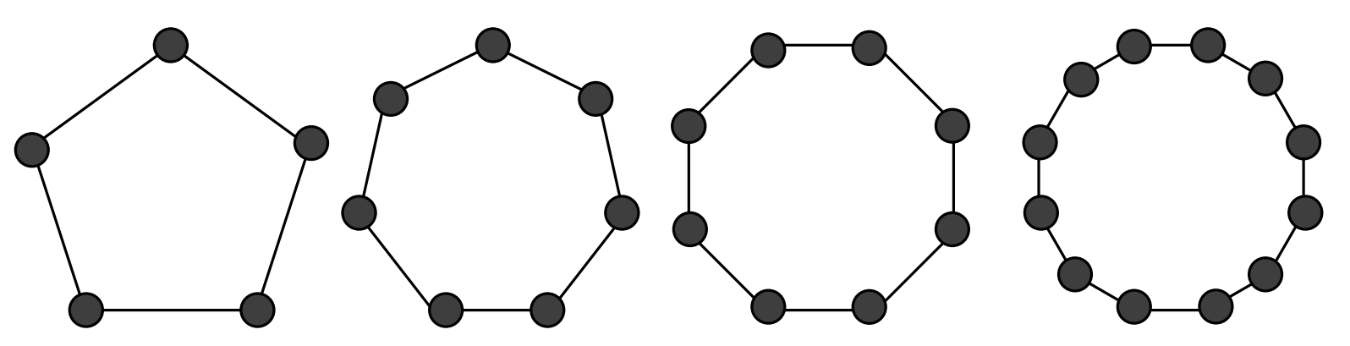
Orders of magnitude faster than homomorphic encryption-based methods

Note: Each chunking round has to use a different communication graph every chunking round.
- Can be done simply by shuffling network addresses (needs network router's help)
- Privacy breach probability can be made negligible by taking a large Nc

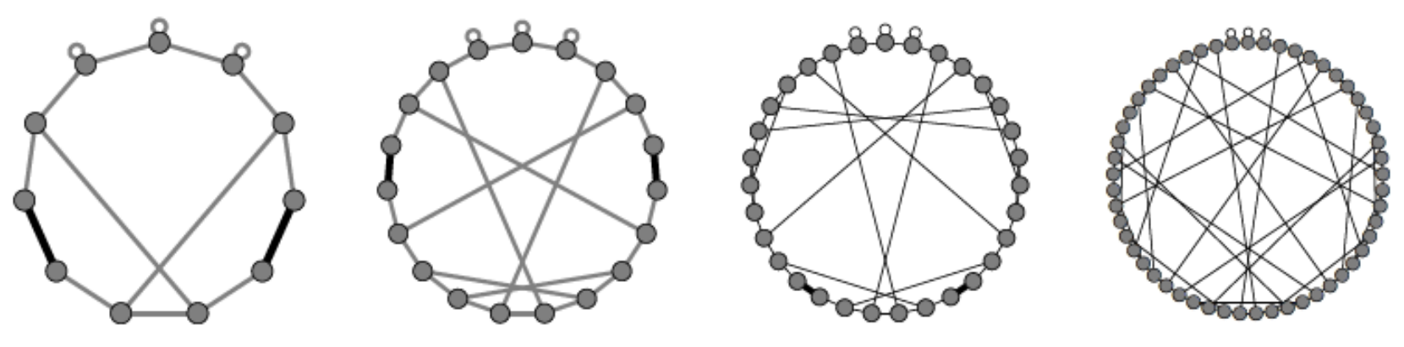
Graph structure matters!

What kind of communication graph *A* should be chosen?

Cycle graph
- Most sparse and symmetric
- Slow convergence (quadratic in S)



"Cycle graph with inverse chord"
- Not regular/symmetric
- FAST convergence (log S)



Motivating application:
collaborative condition-based monitoring of industrial assets

- IoT data is generally noisy
- Data privacy is a major concern but sample-wise encryption is not practical
- Need a new approach!