Efficient Protocol for Collaborative Dictionary Learning in Decentralized Networks

T. Idé ("Ide-san"), R. Raymond, and D. T. Phan
IBM Research, T.J. Watson Research Center
IBM Research - Tokyo
IBM Research

Agenda

- Problem setting and overview
- Global-local separation in exponential family
- Dynamical consensus with privacy
- Numerical example
What is this in a nutshell?  
Multi-task density estimation + a novel set of constraints

- What is multi-task learning?
  - Multiple agents learn their own model based on their own data
  - Leverage commonality among agents to get a better model

- What are the “constraints”?
  - Data privacy
  - Decentralized

This one is tricky
Decentralized learning may be harder than you might think...

- Example: Aggregation (summation)
  \[ \overline{C} = C_1 + C_2 + \ldots + C_6 \]

- Easy? Not really, if we do that only through local communications
  - Broadcasting your data to all?
    - Yes! Total privacy breach
  - Select a leader to let her compute?
    - Yes! What if she is a bad guy?
Problem setting:
Decentralized multi-task density estimation with data privacy

Each agent wishes to learn its own probability density

Closed, membership-based, and decentralized network

Multiple “semi-honest” agents privately keep own data

Real-valued noisy data are assumed

No central server!
Our solution: summary

Maximum likelihood

- Mixture of exponential family
  - For principled probabilistic multi-task learning
- Dynamic consensus on commutation graph
  - For decentralized learning
- Simple secret sharing scheme
  - For data privacy
(For reference) 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

**Decentralized**
- 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 too slow
IBM Research

Agenda

- Problem setting and overview
- Global-local separation in exponential family
- Dynamical consensus with privacy
- Numerical example
We employ a mixture of exponential family for multi-task density estimation

- Each agent holds its own data
  \[ D^s = \{ x^{s(n)} | n = 1, \ldots, N^s; \; x^{s(n)} \in \mathbb{R}^M \} \]

- Employ a mixture model with agent-specific weights
  - \[ p(x | \Theta, \Pi^s) = \sum_{k=1}^{K} \pi_k^s f(x | \theta_k) \]
  - The mixture coefficients \( \{\pi^1, \ldots, \pi^s\} \) is agent-specific
  - \( \{\theta_1, \ldots, \theta_K\} \) are shared by all the agents

- For \( f \), employ exponential family
  \[ f(x | \theta_k) = G(\theta_k) H(x) \exp \{ \eta(\theta_k)^\top T(x) \} \]
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)

Global consensus:
• Compute aggregation
• Perform optimization to store a unique result

Iterates until convergence
IBM Research

Agenda

- Problem setting and overview
- Global-local separation in exponential family
- Dynamical consensus with privacy
- Numerical example
Decentralized aggregation = Finding stationary state of Markovian process

- Consider an aggregation task in general:
  \[
  \bar{c} = \sum_{s=1}^{S} c^{s} = 1^{\top} c
  \]
  S-dimensional vector of ones

- Idea: consider Markovian process whose stationary state is proportional to the \( \mathbf{1} \) vector
  \[
  c^{s} \leftarrow c^{s} + \epsilon \sum_{j=1}^{S} A_{s,j}(c^{j} - c^{s}) \quad \text{or} \quad c \leftarrow [\mathbf{I} - \epsilon(\mathbf{D} - \mathbf{A})]c
  \]
  - \( \mathbf{A} \): Incidence matrix of the communication graph
  - \( \mathbf{D} \): Degree matrix of \( \mathbf{A} \)

- This update equation converges to \( \bar{C} \) in all nodes

Global consensus:
- Compute aggregation
- Perform optimization to store a unique result

The largest eigenvector is \( \mathbf{1} \)
What kind of communication graph $A$ should be chosen?

- **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" (or modified cycle graph)**
  - **Pros**
    - ✔ Sparse.
    - ✔ Fast convergence ($\sim \log S$)
  - **Cons**
    - ✔ Not regular (=not fully democratic)
    - ✔ Eigenvalue is non-smooth with $S$
How can we guarantee data privacy?
Simple secret sharing scheme by random chunking

- Randomly split datum into $N_c$ chunks
  - $\bar{C} = \sum_{s=1}^{S} C^s$

- Do dynamic consensus $N_c$ times and sum
  - $\bar{C} = \bar{C}[1] + \bar{C}[2] + \bar{C}[3]$
  - Note: Each chunking round has to use a different communication graph
    - Can be done simply by shuffling network address (needs network router’s help)
    - Take a large $N_c$ to make the probability of sending all the chunks to the same neighbor negligible
IBM Research

Agenda

- Problem setting and overview
- Global-local separation in exponential family
- Dynamical consensus with privacy
- Numerical example
Much faster than homomorphic encryption-based decentralized approach

- Proposed method is orders of magnitude faster than an existing fully-decentralized consensus algorithm using homomorphic encryption [Ruan+ 17;19]

- Number of iteration is $\sim \log S$ when using the modified cycle graph
  - $S$: The number of nodes

![Figure 5: Computation time comparison between the proposed dynamic random chunking and the encryption-based method.](image)

![Figure 4: The number of iterations $t$ to achieve $\left(\frac{S}{\sqrt{S}}\right)^t = \frac{1}{\sqrt{S}} 10^{-3}$.](image)
Thank you!