Asynchronous **Diffusion** Learning with **Agent Subsampling** and **Local Updates**

Elsa Rizk[1] **Kun Yuan**[2] **Ali H. Sayed**[1]

[1] School of Engineering, École Polytechnique Fédérale de Lausanne

[2] Center for Machine Learning Research, Peking University

About the authors

- Kun Yuan: Assistant Professor @ Center for Machine Learning Research, Peking University
- Research area: Distributed optimization (with decentralized/federated/compressed comm.)

Elsa Rizk

Ali H. Sayed

Collaborative learning with edge devices

- Collaborative learning with edge devices will result in **edge intelligence**

- Each device collects local dataset; small, valuable and sensitive; cannot be shared (data silos)







- Collaborate to train a global model without sharing data (break the data silos)

Data silos in edge computing

[https://m.thepaper.cn/newsDetail_forward_25528954]

Collaborative learning

A network of K agents (nodes) collaborate to solve the distributed optimization problem:

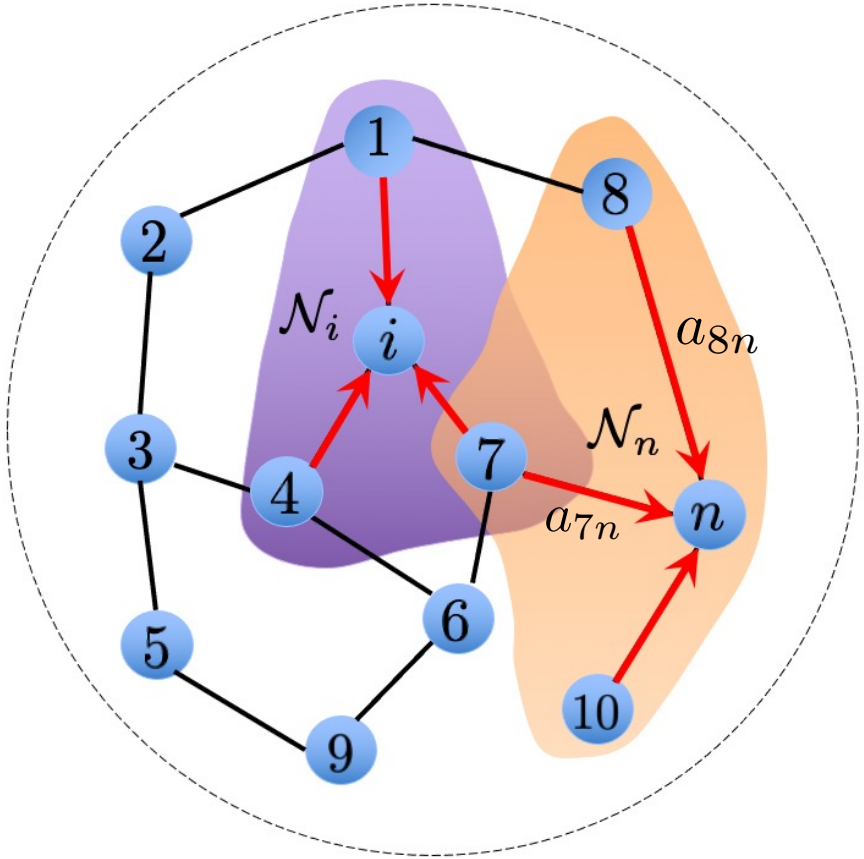
$$\min_{w\in\mathbb{R}^M} \quad \frac{1}{K}\sum_{k=1}^{K} J_k(w) \quad \text{where} \quad \boxed{J_k(w)} = \boxed{\mathbb{E}_{x_k\sim\mathcal{D}_k}[Q_k(w;x_k)]} \tag{1}$$

- Each component $J_k(w):\mathbb{R}^M\to\mathbb{R}$ is local and private to node k
- Random variable x_k denotes local data that follows distribution $\mathcal{D}_k$
- When each distribution $\mathcal{D}_k$ is identical, all $J_k(w)$ share the same solution; otherwise **data heterogeneity** exits

Decentralized collaborative learning

- Assume all agents are organized into an arbitrary connected network
- $\mathcal{N}_k$ is the set of neighbors of node k
- Weight $a_{\ell k}$ is to scale information flowing from node l to node k; If node k is not connected to node l, we have $a_{lk} = 0$
- We introduce combination matrix

$$A = [a_{\ell k}] \in \mathbb{R}^{K \times K}$$

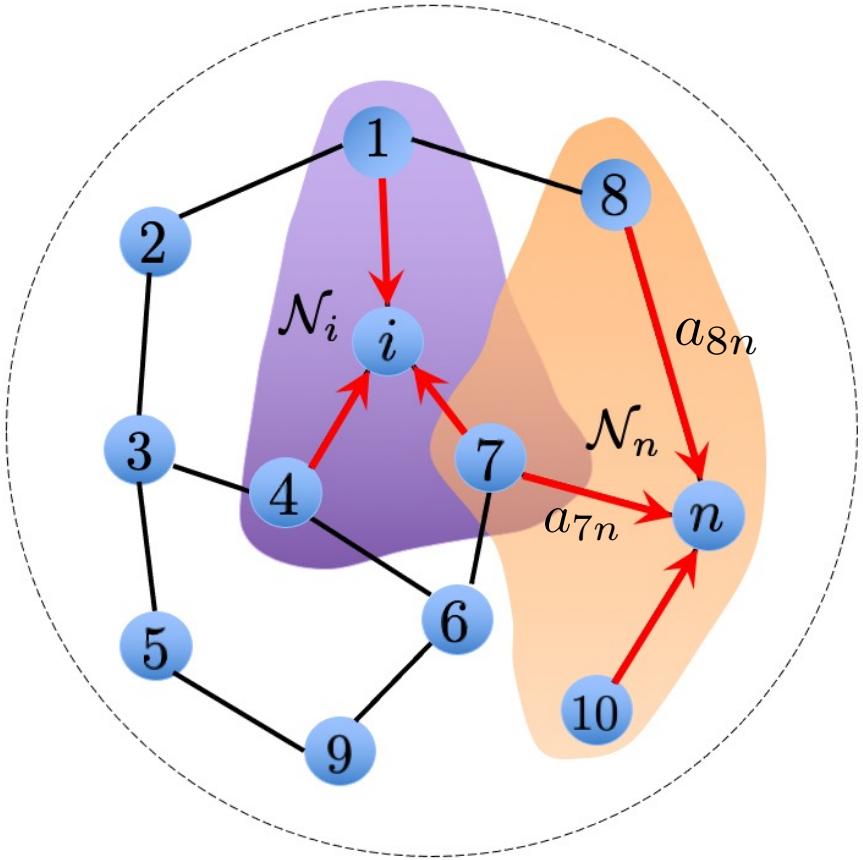


Decentralized collaborative learning

- **Diffusion learning** [Lopes & Sayed, 2009, Chen & Sayed, 2012] (also known as **decentralized SGD**)

$$\psi_{k,i} = w_{k,i-1} - \mu \nabla_w Q(w_{k,i-1}; x_{k,i})$$
$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}$$

- Various variants such as **EXTRA** [Shi et. al., 2015], **Gradient tracking** [Xu et. al., 2015, Lorenzo & Scutari, 2016, Nedic et. al., 2017], and **Exact diffusion** [Yuan et. al., 2019]



Decentralized learning with local update

- In edge computing, **communication is typically very expensive**.
- It is preferred to let each node conduct multiple **local updates** before the communication
- We let the boldface symbol $\boldsymbol{w}$, $\boldsymbol{\psi}$ and $\boldsymbol{x}$ to denote the networked variables

$$\boldsymbol{\psi}_i = \boldsymbol{w}_{i-1} - \mu \nabla_w Q(\boldsymbol{w}_{i-1}; \boldsymbol{x}_i)$$
$$\boldsymbol{w}_i = \mathcal{A}^{\mathsf{T}} \boldsymbol{\psi}_i$$

Diffusion learning

Decentralized learning with local update

- In edge computing, **communication is typically very expensive**.
- It is preferred to let each node conduct multiple **local updates (LU)** before the communication
- We let the boldface symbol $\boldsymbol{w}$, $\boldsymbol{\psi}$ and $\boldsymbol{x}$ to denote the networked variables

$$\boldsymbol{\psi}_{(i-1)T+t} = \boldsymbol{w}_{(i-1)T+t-1} - \mu \nabla_w Q\left(\boldsymbol{w}_{(i-1)T+t-1}; \boldsymbol{x}_{(i-1)T+t}\right)$$

$$\boldsymbol{w}_{(i-1)T+t} = \mathcal{A}^{\mathsf{T}}_{(i-1)T+t} \boldsymbol{\psi}_{(i-1)T+t} \qquad \text{where } t \in \{1, 2, \ldots, T\}$$

$$\text{and } \mathcal{A}_{(i-1)T+t} = \mathcal{A} \text{ if } t = T \text{ otherwise } \mathcal{A}_{(i-1)T+t} = I$$

Diffusion learning with T local updates

Decentralized learning with local update

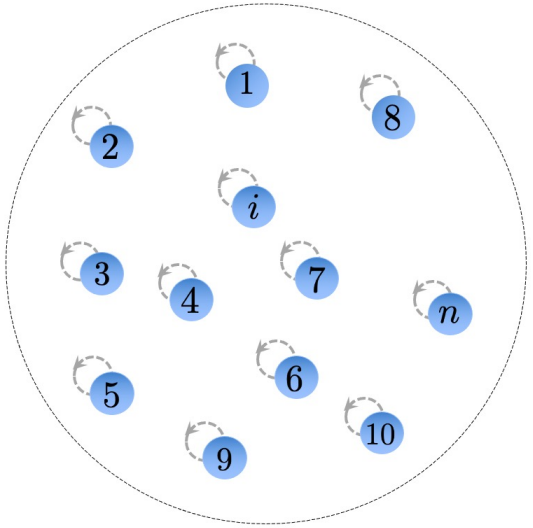
Diffusion learning with T local updates

$$\boldsymbol{\psi}_{(i-1)T+t} = \boldsymbol{w}_{(i-1)T+t-1} - \mu \nabla_w Q(\boldsymbol{w}_{(i-1)T+t-1}; \boldsymbol{x}_{(i-1)T+t})$$

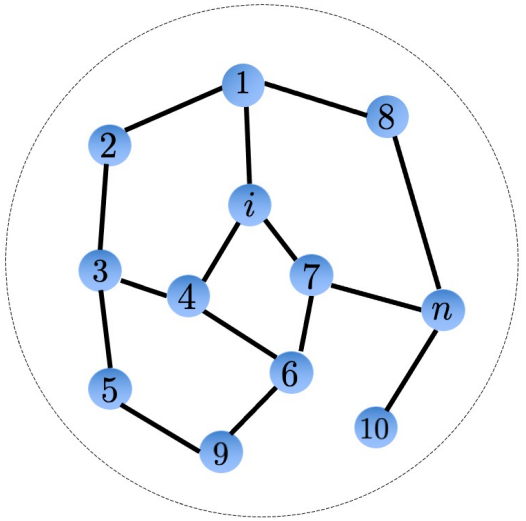
$$\boldsymbol{w}_{(i-1)T+t} = \mathcal{A}^{\mathsf{T}}_{(i-1)T+t} \boldsymbol{\psi}_{(i-1)T+t} \qquad \text{where } t \in \{1, 2, \ldots, T\}$$

$$\text{and } \mathcal{A}_{(i-1)T+t} = \mathcal{A} \text{ if } t = T \text{ otherwise } \mathcal{A}_{(i-1)T+t} = I$$

Shape of the topology



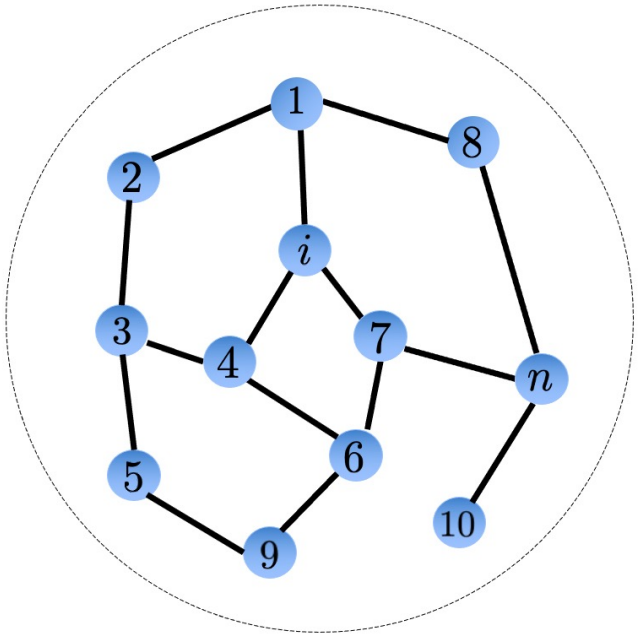
$1 \leq t \leq T-1$



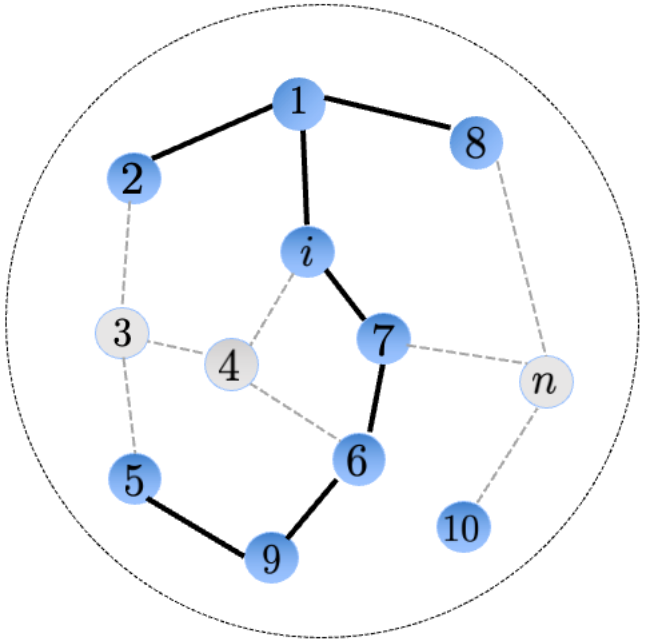
$t = T$

Diffusion learning with LU and agent sampling

- In edge computing, devices are **unstable**; unavailable when out of power of signal
- It cannot be guaranteed that every node can communicate and update as desired
- Only a partial set of nodes is available to join the training process; **agent sampling (AS)**



Ideal scenario



Practical scenario

Unavailable node

Unavailable link due to unavailable node

Diffusion learning with LU and AS

- To model partial sampling in diffusion learning, assume each node is activated independently at

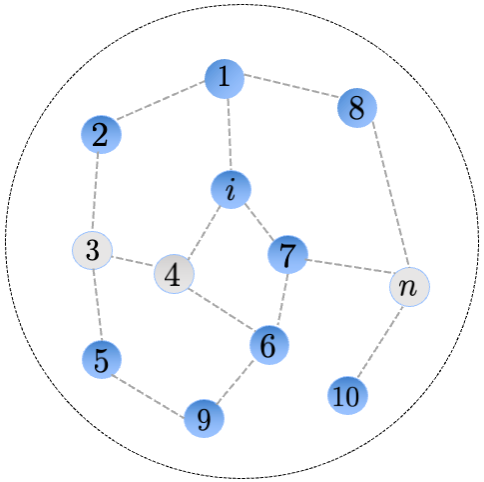
$$\Pr(\text{node } k \text{ is available at iteration } i) = p_k$$

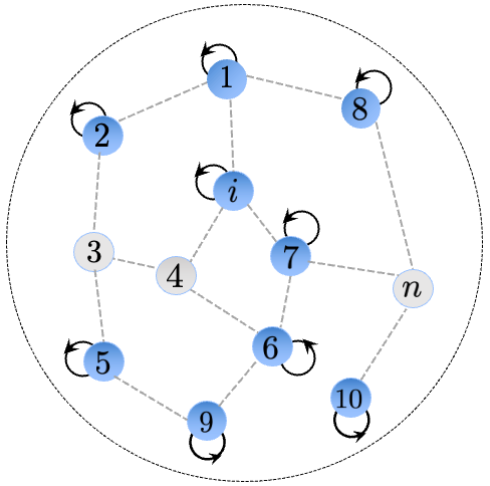
(a) At the beginning of each outer iteration i, a set of nodes are available

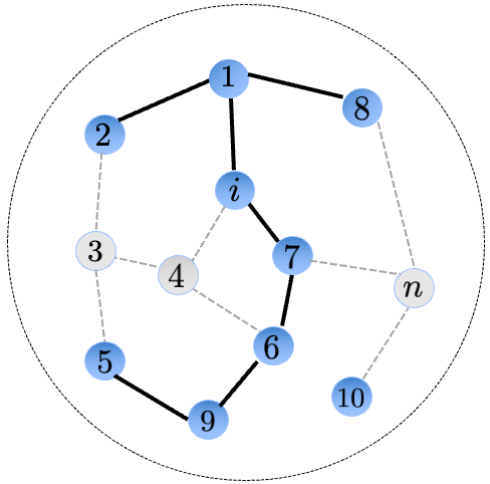
(b) After their activation, these nodes start performing T local updates

(c) After T local updates, these nodes start communicating following the active links

repeat







Diffusion learning with LU and AS

Since the network topology has been changed, we have to **reweight the topology** $\mathcal{A}_{iT}$

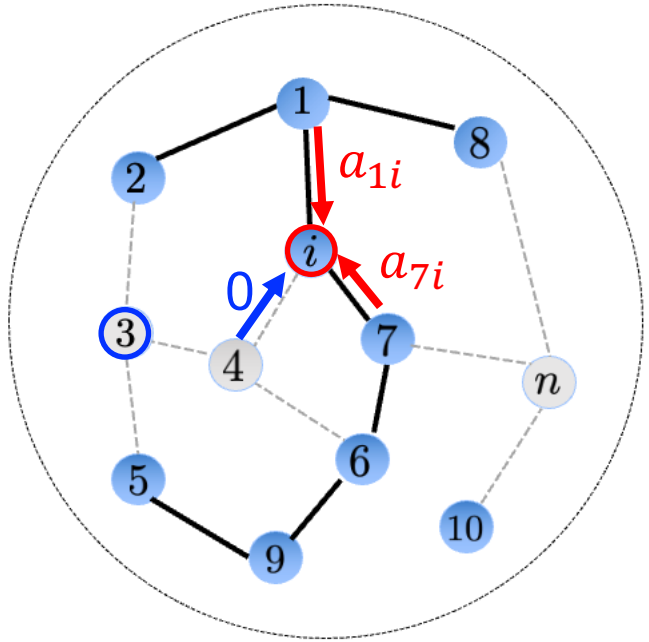
- All active agents send variables to neighbors
- If active agent k receives variables from agent l

$$a_{\ell k,iT} = a_{\ell k}$$

- If active agent k does not receive variable from agent l

$$a_{\ell k,iT} = 0$$

- Active agent k sets its own weight as $a_{kk,iT} = 1 - \sum_{l \in N_k} a_{lk,iT}$



- Inactive agent k sets its own weight as $a_{kk,iT} = 1$ and $a_{lk,iT} = 0$ for any $\ell \neq k$

Diffusion learning with LU and AS

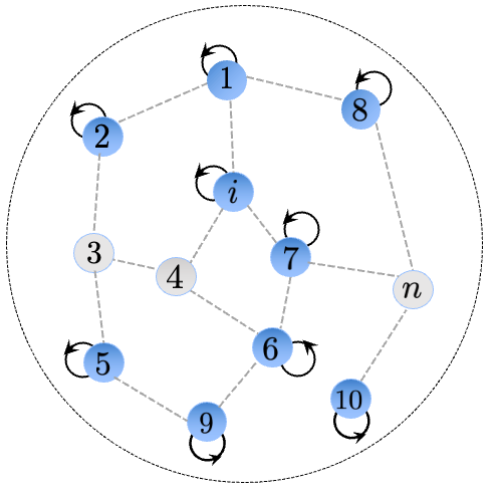
Diffusion learning algorithm with both local updates and agent sampling

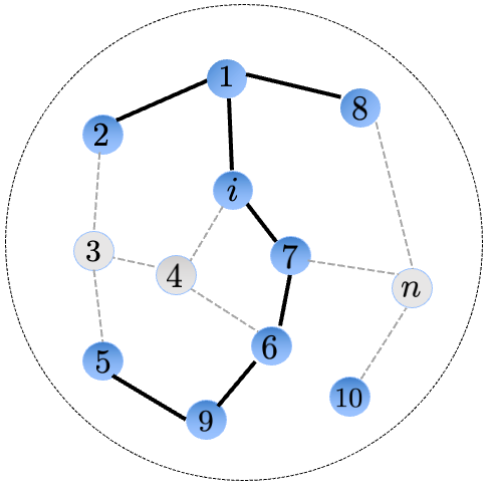
$$\psi_{k,(i-1)T+t} = w_{k,(i-1)T+t-1} - \mu_{k,(i-1)T+t} \nabla_w Q(w_{k,(i-1)T+t-1}; x_{k,(i-1)T+t})$$

$$w_{k,(i-1)T+t} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k,(i-1)T+t} \psi_{\ell,(i-1)T+t} \qquad \text{where } t \in \{1, 2, \ldots, T\}$$

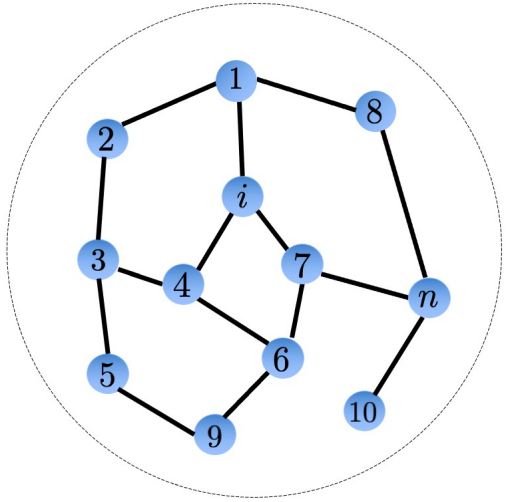
and $\mu_{k,(i-1)T+t} = \mu$ if agent i is active otherwise $\mu_{k,(i-1)T+t} = 0$

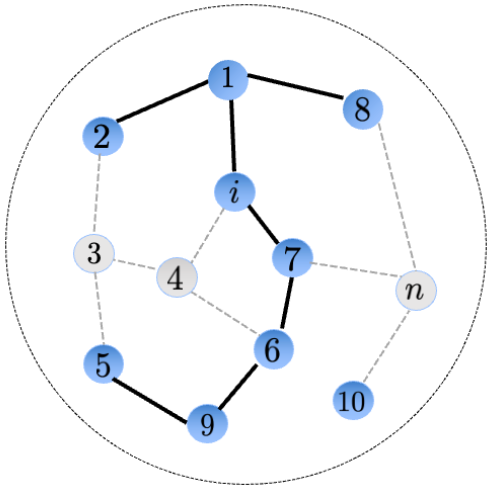
and $\mathcal{A}_{(i-1)T+t}$ will be set as in Page 12 if $t = T$ otherwise $\mathcal{A}_{(i-1)T+t} = I$





Reweighted combination matrix





- The underlying matrix A is **symmetric** and **doubly-stochastic**

  Reweighting matrix A_{iT} is also **symmetric** and **doubly-stochastic**

- The underlying matrix A is **doubly-stochastic**

  Reweighting matrix A_{iT} is **NOT** necessarily **doubly-stochastic**

  But it is guaranteed to be **left-stochastic**

- The underlying matrix A is **left-stochastic**

  Reweighting matrix A_{iT} is **left-stochastic**

Convergence analysis

Assumption 1 [Combination matrix]

The combination matrix A is symmetric and doubly-stochastic, i.e.,

$$A = A^{\intercal} \quad A\mathbf{1} = \mathbf{1}$$

Assumption 2 [Loss function]

Each $Q_k(w; x_k)$ is convex in terms of x and δ-smooth. Furthermore, we assume $J_k(w)$ is ν-strongly-convex, i.e.

$$\|\nabla_w Q_k(w_1; x_k) - \nabla_w Q_k(w_2; x_k)\| \leq \delta \|w_1 - w_2\|$$

$$J_k(w_2) \geq J_k(w_1) + \langle \nabla J_k(w_1), w_2 - w_1 \rangle + \frac{\nu}{2}\|w_2 - w_1\|^2$$

Convergence analysis

Assumption 3 [Gradient noise]

We assume the gradient noise is unbiased and has bounded variance, i.e.

$$\mathbb{E}[\nabla_w Q_k(w; x_k)] = \nabla J_k(w)$$
$$\mathbb{E}\|\nabla_w Q_k(w; x_k) - \nabla J_k(w)\|^2 \leq \sigma^2$$

Assumption 4 [Smooth Hessian]

Each $J_k(w)$ is twice-differentible, and its Hessian is locally Lipschitz in a small neighbourhood around $w^\star$ where $w^\star$ is the solution to problem (1), i.e.,

$$\|\nabla_w^2 J_k(w^\star + \Delta w) - \nabla^2 J_k(w^\star)\| \leq \kappa \|\Delta w\|$$

Convergence analysis

Theorem 1 [Convergence stability]

Under Assumptions 1-4, when step-size $\mu \leq 2\delta/\nu^2$, it holds that

$$\limsup_{i\to\infty} \mathbb{E}\|w_{k,(i-1)T+t} - w^\star\|^2 = \mathcal{O}(\mu\sigma^2)$$

Key challenge: **Time-varying** topology; step-size and combination matrix are **dependent** ; **data-heterogeneity** exists

This theorem implies that diffusion learning with both local updates and partial sampling will converge to the solution when the underlying combination matrix is both **symmetric** and **doubly-stochastic**

This theorem only provides the **order** of the convergence error. With more assumptions, we can establish the **precise** mean-square-deviation (MSD) expression

Convergence analysis

Theorem 2 (Steady-state MSD). *It holds that:*

$$\limsup_{i\to\infty} \mathbb{E}\|w_{k,(i-1)T+t} - w^\star\|^2 = \frac{1}{K} z^{\mathsf{T}} \mathrm{bvec}(I) + O(\mu^{1+\alpha_0}), \tag{37}$$

where the block vectorization operator bvec *stacks the columns of the blocks of the matrix, and:*

$$\alpha_0 \triangleq \frac{1}{2}\min\{1, \alpha_s\}, \tag{38}$$

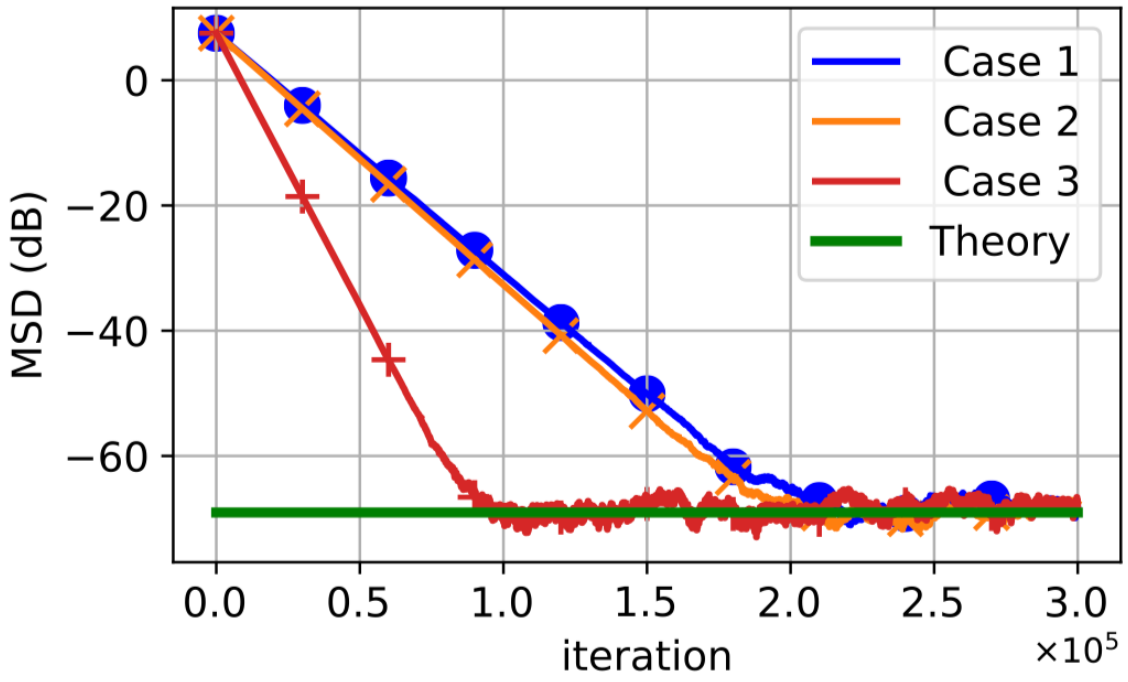
$$z \triangleq \left(I - \left(\mathcal{G}_T \mathcal{G}_t^{T-1}\right)^{\mathsf{T}}\right)^{-1} \mathcal{C}\, \mathrm{bvec}(\mathrm{diag}\{R_k\}), \tag{39}$$

$$\mathcal{C} \triangleq \left(I + \left(\mathcal{G}_T \mathcal{G}_t^{T-1}\right)^{\mathsf{T}}\right) \mathcal{C}_T + \mathcal{G}_T^{\mathsf{T}} \sum_{j=1}^{T-1} \left(\mathcal{G}_t^{j-1}\right)^{\mathsf{T}} \mathcal{C}_t. \tag{40}$$

Precise error without any approximation

Numerical experiments

Linear regression $\min_{w} \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{d}_k(n)-\boldsymbol{u}_{k,n}^{\mathsf{T}}w\|^2.$



Case 1: K=20; T=100; $p_k = 0.5$

Case 2: K=20; T=1; $p_k = 0.5$

Case 3: K=20; T=1; $p_k = 1$

Theory: Our established MSD expression

All scenarios match with our MSD expression

Extension

- The above analysis assumes **symmetric** and **doubly stochastic** matrix A
- Can the algorithm converge to the desired solution $w^\star$ **without** symmetricity?
- In this scenario, the time-varying matrix A_{iT} is **left-stochastic**, but not doubly-stochastic
- By assuming **homogeneous** local loss functions, we can achieve the same result

$$\limsup_{i\to\infty} \mathbb{E}\|w_{k,(i-1)T+t} - w^\star\|^2 = \mathcal{O}(\mu\sigma^2)$$

$$\limsup_{i\to\infty} \mathbb{E}\|w_{k,(i-1)T+t} - w^\star\|^2 = \frac{1}{K} z^{\mathsf{T}} \mathrm{bvec}(I) + O(\mu^{1+\alpha_0}),$$

Can left-stochastic + heterogeneity work?

- It is very challenging when data heterogeneity exists when the combination matrix is time-varying and left-stochastic

- Consider the very simple average consensus algorithm with time-varying and left-stochastic matrix A_i

$$x_i = A_i^{\mathsf{T}} x_{i-1} \quad \text{where } A_i \text{ is left-stochastic}$$

- Suppose each A_i is associated with a Perron eigenvector $\pi_i \in R^K$. Average consensus is essentially approximate of the following recursion

$$x_i = \mathbf{1}\pi_i^{\mathsf{T}} x_{i-1}$$

- Since π_i is time-varying, we cannot predict the trend of variable x_i, which brings the major challenge for the algorithm development (Push-sum cannot work)

Summary

- In edge computing, communication is **expensive** and agents are **unstable**
- We develop a diffusion learning algorithm with both **local updates** and **agent sampling**
- Agent sampling will reweight the combination matrix per period, which is one of the major challenges in algorithm development and analysis
- When the underlying matrix is symmetric and doubly-stochastic, we establish convergence guarantees even when data heterogeneity exists
- When symmetric does not hold, convergence can be guaranteed with homogeneous data

EPFL 北京大学 PEKING UNIVERSITY

Thank you!

Kun Yuan homepage: https://kunyuan827.github.io/