POSTER: DYNAMIC DECENTRALIZED FEDERATED LEARNING

Shenghong Dai [1] **Kangwook Lee** [1] **Suman Banerjee** [1]

ABSTRACT

The concern about mobile devices' data privacy has motivated Federated Learning (FL) where many clients (which are coordinated by a central server) collaboratively train a model while communicating only model updates. However, a bottleneck might occur over the central server as the number of clients increases. This limitation motivates existing research for decentralized FL where clients share model updates with their neighbours instead of the central coordinator. We present a decentralized FL algorithm with convergence guarantees that addresses the practical challenge of the time-varying connectivity graph (e.g., neighbors are not fixed).

1 INTRODUCTION

To protect mobile clients' privacy, Federated Learning (FL) as shown in the left of Figure 1 is applied to train a shared model. The centralized FL repeats broadcast, client computation, and aggregation steps until training is stopped. However, a bottleneck might occur over the central server when operating at a larger scale. Moreover, a reliable central server may not always exist. These limitations motivate the need for decentralized FL.

Decentralized FL replaces primitive communication with peer-to-peer communication where client exchanges information with their neighbors. While distributed optimization has been studied in the control community, decentralized SGD has been less explored in machine learning. Fortunately, Lian (2017) has validated the promising future of decentralized SGD algorithms over centralized algorithms.

Further, mobile clients, which may expect large variations in location, might have time-varying neighbors and thus leading to a dynamic connectivity graph. However, little research has considered the decentralized FL in a mobile environment (shown in the right of Figure 1). Therefore, we deploy a dynamic decentralized FL algorithm for the time-varying decentralized network and obtain theoretical convergence bounds of its performance.

2 METHOD

This section provides the dynamic decentralized FL algorithm and then analyzes its convergence rate.

[1]University of Wisconsin-Madison, USA. Correspondence to: Shenghong Dai <sdai37@wisc.edu>.

Proceedings of the 5^{th} MLSys Conference, Santa Clara, CA, USA, 2022.

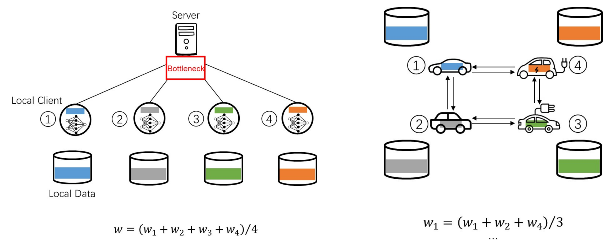

Figure 1. An illustration of centralized FL and decentralized FL with dynamic peer-to-peer communication topology.

2.1 Dynamic Decentralized FL Algorithm

Given an undirected graph G with $V := \{1, 2, \cdots, n\}$ representing the set of clients, $N_i(k) \subset \{1, 2, \cdots, n\}$ denotes the neighbors of the node i in the kth iteration. Then the decentralized communication matrix $E \in R^{n\times n}$ is defined as in (Ram et al., 2009):

$$E_{ij}(k) = \begin{cases} 1/n & j \neq i \text{ and } j \in N_i(k) \\ 1 - |N_i(k)|/n & j = i \\ 0 & \text{otherwise} \end{cases}$$

Beyond an equal contribution, E_{ij} shows how much influence client j will have on client i. Here E is a doubly stochastic matrix since all entries are nonnegative ($E_{ij} \in [0,1], \forall i, j$); the sum of each row is 1 ($\sum_j E_{ij} = 1, \forall i$); and E is symmetric.

Algorithm 1 shows the operation of the ith client under the dynamic decentralized FL algorithm.

2.2 Convergence Rate Analysis

To provide the convergence results, we first state the convergence of non-homogeneous Markov chains which has been proved in (Nedic et al., 2008). They make two assumptions as follows:

Algorithm 1 Dynamic Decentralized FL on the ith Client

Input: initial point $w_{0,i} = w_0$, communication matrix E, the number of iterations K

for $k = 0$ **to** $K - 1$ **do**

Randomly sample data

Update the local model $w_{k,i}$ using the sampled data

Average the local model with neighbours based on the communication matrix: $w_{k+1,i} = \sum_{j=1}^{n} E_{ij}(k) w_{k,j}$

end for

1. There exists an integer $M \geq 1$ such that for all t, the graph $(V, \cup_{k=t}^{t+M-1} \epsilon(E(k)))$ is strongly connected, where $\epsilon(E(k))$ is a set of edges (i, j) such that $E_{ij}(k) > 0$.
2. For all k, if $E_{ij}(k) > 0$ then $E_{ij}(k) > \eta$.

These assumptions guarantee that the clients are connected frequently enough in time, and these lead to the following Lemma:

Lemma 2.1 *Under Assumption 1 & 2, for all t, l with $t \geq l \geq 0$ we have*

$$\left| [E(t)E(t-1)\cdots E(l)]_{ij} - \frac{1}{n} \right| \leq \left(1 - \frac{\eta}{4n^2}\right)^{\frac{t-l}{M}-2}$$

Then Lemma 2.1 can be invoked in the following analysis of (Lian et al., 2017) while keeping its existing assumptions (lipschitzian gradient, spectral gap, bounded variance & start from 0). In all, they can give us the subsequent convergence results of Algorithm 1.

Theorem 2.2 *Under assumption 1, 2 and assumptions in (Lian et al., 2017), if the number of iterations K is large enough and the learning rate is wisely set, Algorithm 1 has the same convergence rate as centralized PSGD:*

$$\frac{\sum_{k=0}^{K-1} E\|\nabla f(\frac{1}{n}\sum_{i=0}^{n-1} w_{k,i})\|^2}{K} \leq \mathcal{O}\left(\frac{1}{K} + \frac{1}{\sqrt{nK}}\right)$$

3 EXPERIMENTS

We implemented the dynamic decentralized FL algorithm 1 on two benchmark MNIST and CIFAR10 datasets to verify our theory; we train a simple CNN model and ResNet56 model from 10 and 30 clients respectively. Although the theoretical convergence guarantee is for one local update per iteration, we relax this assumption in the experiments to 5 local epochs. In addition, we partition data in two different ways: IID, where dataset was randomly partitioned over clients; Non-IID, where each client will have examples of only one MNIST digit or get allocated CIFAR10 using latent dirichlet allocation.

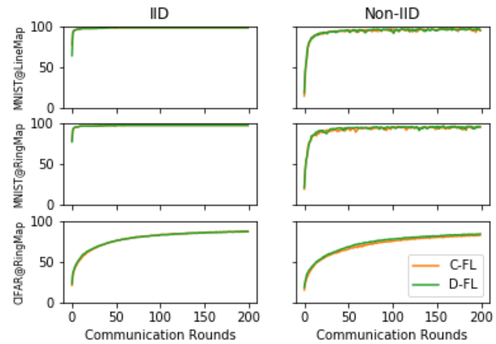

Figure 2. Test Accuracy in various settings.

These experiments consider two special maps: a line with five cities of uniform spacing; a unit circle with ten evenly distributed cities. Specifically, the motion of a mobile client is inversely proportional to the distance between pairs of cities. And the clients in the two adjacent cities are considered neighbors. Test accuracy in Figure 2 validate our theory since D-FL has the same convergence rate.

4 FUTURE WORK

FL researchers typically experiment with some specific datasets that will then be partitioned over clients. However, these datasets may be unreliable. One may generate a dataset from virtual agents who can collect data in a virtual world. Specifically, one could take advantage of the open-source simulator (e.g., GTA V) to collect decentralized data. Compared with the pre-existing FL datasets, this dataset better conforms to the actual data because it is generated from mobile nodes. Besides, this dataset is naturally partitioned, thus minimizing discrepancies due to manually designed partition methods. Furthermore, the simulator provides a realistic modeling for the dynamic graph.

REFERENCES

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

Nedic, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. N. Distributed subgradient methods and quantization effects. In *2008 47th IEEE Conference on Decision and Control*, pp. 4177–4184. IEEE, 2008.

Ram, S. S., Nedić, A., and Veeravalli, V. V. Incremental stochastic subgradient algorithms for convex optimization. *SIAM Journal on Optimization*, 20(2):691–717, 2009.