Iterative Improvement on Different Topologies under the Effect of Heterogeneous Data with Decentralized SGD

Chun Hei Michael Chan, Bohan Wang, Qi Yi
Mini-project for CS-439 Optimization for Machine Learning

***Abstract*—Decentralized learning is faced with a great challenge that most algorithms fail to efficiently deal with the heterogeneity of data across nodes, which has a coupling effect on the graph communication topology as well. To reveal the effects of different topologies on decentralized SGD (D-SGD), we firstly compare the convergence performance on eight typical topologies with iid and non-iid data. Secondly, an iterative algorithm is designed to improve the ring and binary-tree topology with heterogeneous data distributions. Finally, our experiments compare the refined topologies with the baseline, providing some valuable insights on improving different topologies.**

I. Introduction

Data heterogeneity is widely regarded as one of the key challenges in decentralized learning [1]. Since the i.i.d. assumption of data is not realistic for decentralized learning. For instance, federated learning, which requires data privacy, is assumed to have training data that is heterogeneous on different nodes. The learning task is distributed over n machines that can only communicate to their neighbors on a fixed communication graph. And the decentralized topology setting is applied to avoid the bottlenecks on the central node in terms of communication latency, bandwidth and fault tolerance.

Recent studies suggest an interplay between the topology and the data distribution across nodes. The effect of the topology can be ignored for the homogeneous data [2]. Nevertheless,in the heterogeneous setting, the choice of topology has a large influence on the convergence performance of decentralized learning [3].

The effect of heterogeneous data on different topology still remains entirely empirical and needs to be technically understood. In this report, firstly the effect of different topologies on the convergence of decentralized SGD (D-SGD) [4] with homogeneous and heterogeneous data are compared, providing a good choice of topology regarding heterogeneity. Secondly, we developed an iterative algorithm to refine the connectivity of different topologies by greedily adding new edges, which efficiently improve the generality of D-SGD. Finally, the ring topology and binary-tree topology are chosen as the standard topology to perform the iterative algorithm, and the results are compared with the baseline, offering some interesting findings.

II. MODELS AND METHODS

A. Problem model

In decentralized learning, we consider optimization problems distributed across n devices or nodes of the form

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^d} \left[f(\mathrm{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \right] \tag{1}$$

where $f_i = \mathbb{R}^d \to \mathbb{R}$ for $i \in [n] := 1, ..., n$ are the objectives defined by the local data available on each node. In machine learning applications, this corresponds to minimizing an empirical loss f averaged over all local losses f_i.

B. Heterogeneous data

Consider a set $N = 1, ..., n$ of n nodes seeking to collaboratively solve a classification task with K classes, We denote a labeled data point by a tuple of random variables $(X_i, Y_i) \sim \mathcal{D}_i$ where $X_i \in \mathbb{R}^q$ represents feature vector and $Y_i \in [\![1, ..., K]\!]$ for its label, and the local distribution D_i on node i may differ from that of other nodes. Essentially, the heterogeneity of the distributions $\{D_i\}_{i=1}^n$ comes from a difference in the label distribution $P_i(Y)$, known as label skew, a important type of data heterogeneity in federated classification problems [5], defined with:

$$\forall i \in [\![1, ..., n]\!], \mathcal{D}_i = P_i(X, Y) = P(X|Y)P_i(Y) \tag{2}$$

This means that the probability $P_i(X, Y)$ under the local distribution $\mathcal{D}_i$ of node i, decomposes as $P_i(X, Y) = P(X|Y)P_i(Y)$, where $P_i(Y)$ may vary across nodes.

C. Topology setting

The network topology is modeled as a graph $G = [(n], E)$ edges $i, j \in E$ if and only if nodes i and j are connected by a communication link, meaning that these nodes directly can exchange messages. In addition we define a doubly stochastic and symmetric mixing matrix $W \in \mathbb{R}^{n\times n}$ as a weighted adjacency matrix of G with the weights $w_{ij} \in [0, 1], w_{ij} > 0$, iff $(i, j) \in E$ and the matrix is doubly stochastic $\sum_{i=1}^n w_{ij} = 1$

To reveal the effects of non-iid data, 5 classical types of topologies are chosen (the topologies shown in Appendix 5), which includes binary tree, torus, ring, chain, and fully-connected. In addition, we extend the existing approach by using three topologies which are ladder, barbell and wheel.

D. Iterative algorithm of improving topology

1) structure of D-SGD: In D-SGD, every worker $i \in [n]$ maintains local parameters $x_i^{(t)} \in R^d$ that are updated in each iteration with a stochastic gradient update (computed on the local function f_i) and by averaging with neighbors in the communication graph. And the mixing step is denoted as multiplication with the mixing matrix W follow the topology setting. The whole process of D-SGD can be illustrated as appendix 2.

2) Improving topology dynamically based on iteration: It has been shown that the impact of data heterogeneity can be significantly reduced by carefully designing the underlying communication structure in [3]. Inspired by their work, we designed an iterative algorithm 1 to refine the structure of topology, by adding new edges via a heuristic way based on the accuracy and data of each node. Distinguished from the previous studies, the mixing matrix remains unchanged (keeping the value of each node uniformly distributed), and perform adjustment on the connectivity of edges in each iteration.

Algorithm 1 ITERATIVE TOPOLOGY UPDATE

```
Input: $X$ dataset, number of iterations $T$, number of edges
    to add $K$, topologies $P_k$, number of nodes $N$
 1: $P_0$ := starting topology (e.g Ring)          ▷ set a topology
 2: for $k$ in $0, ..., K$ do
 3:     for $t$ in $0, ..., T$ do
 4:         for $i$ in $0, ..., N$ do
 5:             $M_i$ = DSGD($P_k$, X)              ▷ models for node i
 6:             $acc_i^{val} = M_i(X)$               ▷ validation accuracy
 7:     $m = argmin_{i \in nodes}(acc_i^{val})$      ▷ select first node m
 8:     Find neighbor nodes of $m$ as $m^{nei}$
 9:     $L_m = Label_m$                  ▷ record label set of node $m$
10:     $L_m^{nei} = Label_{m^{nei}}$      ▷ label set of neighbors of $m$
11:     $n = argmax_{j \neq m, j \notin m^{nei}}(L_j - (L_m^{nei} \cup L_m))$
12:     Add edge $E_{mn}$ in to the topology $P_0$
13:     Reset weights of all models $M_i$
```

To efficiently improve the topology, in line 7 first we select the node m with the lowest validation accuracy, and then try to find another node n to be attached, the reason why using validation accuracy is to improve the generality of the node model. In line 11 we choose the destination node n following two simple principles: 1) node n should not be the neighbors of node m or itself. 2) node n holds the most non-overlapping labels with node m's neighbors (as our mixing matrix does not have self loops). The underlying idea of this heuristic method is to refine the connectivity between these heterogeneous nodes, making the node-level label distribution closer to the global distribution so as to improve the convergence. In line 17, we reset weights so to compare all topologies from scratch, but a possible variant would be to keep the weights intact.

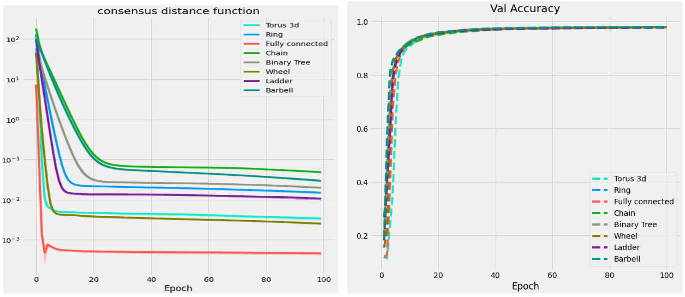
III. Experiments

In this work, we use convolutional neural networks (CNNs) trained on MNIST, which has a training set of 60000 examples with 10 output classes [6]. We spilt the dataset into 40000 training sets, 10000 testing sets and 10000 validation sets. The parameters for training are: batch size $B = 100$, number of epochs $E = 100$, learning rate $r = 5e-4$, and $momentum = 0.9$.

The training sets are evenly partitioned into 8 nodes, where each topology has the same number of nodes. For IID setting, each node is randomly assigned a uniform distribution over 10 classes. For non-IID setting, we create two extreme cases: (a) one label skewed non-IID, where each client is randomly assigned with 4000 images from one class. (b) two labels skewed non-IID, where we first sort the data by digit label, divide it into 20 shards of size 2000, and assign each node 2 shards create 2-class label skew [7], which is a pathological case where each node holds examples from only two classes.

A. Convergence of D-SGD on different topologies

To show why topology matters in decentralized optimization, firstly we compare the convergence performances on different topologies with iid dataset in Fig. 1. We set the maximum number of epoch as 100 and each training was run 10 times to compensate for the stochastic error. The results demonstrate that, fully connected topology and torus topology can help D-SGD converge fast. By comparison, D-SGD with chain and barbell topology converges the slowliest. On the other side, all topologies show similar convergence trends in each epoch in terms of validation accuracy. This conclusion is also exactly in line with previous observations in Neglia's work [8], i.e. the topology does not matter with iid data in D-SGD. To summarize, torus and ring topology are the best topology for iid data in terms of the communication efficiency, the convergence rate and generality.

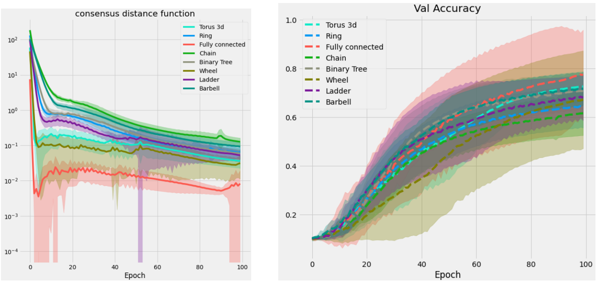


(a) Consensus distance function (b) Validation accuracy

Fig. 1. Convergence of D-SGD with different topologies on iid data

For the two labels skewed non-iid data, the convergence curves of D-SGD with different topologies are affected. It is obvious that all the convergence curves fluctuate largely in Fig. 2 (a). In particular, the Ladder topology is influenced the most. However, the convergence rates of D-SGD with different topologies remain similar to that of the iid setting. Significant reduction in the validation accuracy is observed for D-SGD on non-iid data (Fig. 2 (b)). As shown in Fig. 2, fully connected topology (28 edges) helps D-SGD on non-iid data achieve the best validation performance (at about 80% accuracy), which is meaningful because all nodes in fully connected graph can receive enough information. Torus topology (12 edges), barbell topology (13 edges) and binary-tree topology (7 edges) can also achieve relatively good validation performance (reaching about 75% accuracy). On the other side, the validation accuracy of D-SGD with ring topology (8 edges) and chain topology (7 edges) is reduced the most by non-iid data.

To summarize, there is a trade-off between graph communication efficiency and the generality of D-SGD. Empirically, the topology containing more edges can achieve better generalization. This is verified by our experiment, because fully connected topology achieves the best validation performance. However, binary-tree containing only 7 edges achieves similar generalization on different distributions. In the following experiments, we implement our designed algorithm to improve the performance of binary-tree further. We also improve the ring topology, because ring topology is communication efficient, but achieves the worst generalization.



(a) Consensus distance function (b) Validation accuracy

Fig. 2. Convergence of D-SGD with different topologies on non-iid data.

B. Refine convergence and topology dynamically under Heterogeneous Data

In this section we observe results around the updated topologies (with different starting topologies). Various experiments were done to test improvements involved.

1) First Experiment: To start with, we apply the algorithm over 10 distributions . We look at Fig. 3 and notice that most topologies with added edges consistently perform better in validation accuracy (computed by doing the mean over the nodes) than the standard topology while keeping early on accuracy just as high, despite adding edges. In more details, the evolution of a selected node's accuracy seems as well positively caused by our edge attaching step 7. These results are in line with what we expected since, having connectivity would allow for each node to receive a larger sample/representation of each labels and thus should improve overall accuracy.

2) Second Experiment: In a second step, given the first experiment, we select a modified topology (+ 5 edges) and a fixed distribution and this time around, apply 10 runs on the training process itself. We see in Fig. 4 where for either ring or binary tree starting topology, adding 5 extra edges gives us in average better converged accuracy (more detailed in Fig. 8) , and weights consensus distances are converging to 0 faster. An interesting point is that, when compared to Torus (12 edges) which originally performs better than Ring Fig. 2 while having lower accuracy early on, the refined Ring has similar final accuracy and seems to have higher accuracy all along training epochs on validation set, the error bar does not give us full confidence in this interpretation however, but this might be one advantage in the iterative refining we attempt. Lastly, with these 10 runs we in parallel evaluate the test accuracy for the topology selected and show them against its standard D-SGD (non iid) counter parts (Table. I). We expected a better test accuracy from refined Ring (Algorithm 1) due to the choice of edges added to be such that it improves on the supposed weakest node model each time, and so we hoped that adding weak knowledge of the non-iid distribution would perform well. However the results show a rather similar test accuracy for both topologies of similar edge count.

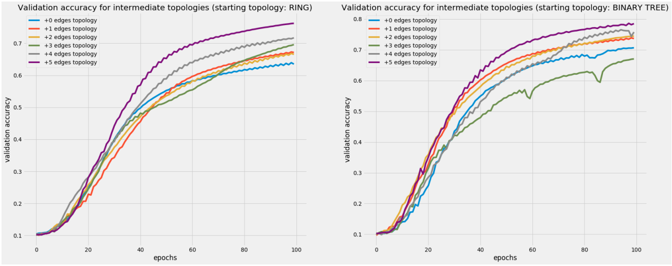


Fig. 3. Validation accuracy on intermediate topologies on non-iid datasets for different starting topology

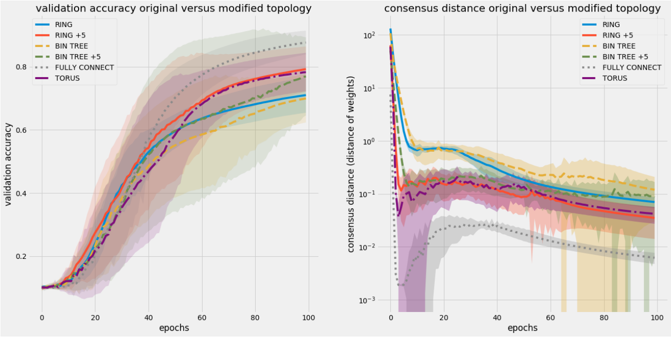


Fig. 4. Validation accuracy on same distribution for 10 training runs comparing selected modified topology and its starting topology

IV. Conclusion

From the conducted experiments we summarize the following findings. In the context of iid distribution, the topology has minimal influence, whereas in non-iid situation the obtained accuracies largely depend on the topology used. We explore an attempt on constructing a small graph starting from small edge count standard graphs. Adding edges to topologies improves the general accuracy of the models, however the topology we obtain by adding edges does not yield a significantly higher accuracy than other standard graphs of same edge counts.

Methods/Topologies	Ring	Binary Tree	Torus	Fully connected
D-SGD (iid)	0.983 ± 0.001	0.983 ± 0.001	0.983 ± 0.001	0.983 ± 0.001
D-SGD (non iid)	0.655 ± 0.088	0.717 ± 0.074	0.737 ± 0.066	$\mathbf{0.783 \pm 0.188}$
ALGO 1 (non iid)	0.776 ± 0.070	0.797 ± 0.073	-	-
D-SGD (fixed)	0.728 ± 0.056	0.713 ± 0.077	0.796 ± 0.060	0.890 ± 0.035
ALGO 1 (fixed)	0.809 ± 0.071	0.782 ± 0.124	-	-

TABLE I
TEST ACCURACY OF DIFFERENT TOPOLOGIES AND DIFFERENT METHODS. NOTE THE FOUR TOPOLOGIES IN THE TABLE ACHIEVE THE BEST TEST PERFORMANCE IN SEC. III-A.

References

[1] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," *CoRR*, vol. abs/1907.09356, 2019. [Online]. Available: http://arxiv.org/abs/1907.09356

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

[3] A. Bellet, A. Kermarrec, and E. Lavoie, "D-cliques: Compensating noniidness in decentralized federated learning with topology," *CoRR*, vol. abs/2104.07365, 2021. [Online]. Available: https://arxiv.org/abs/2104.07365

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

[5] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4387–4398.

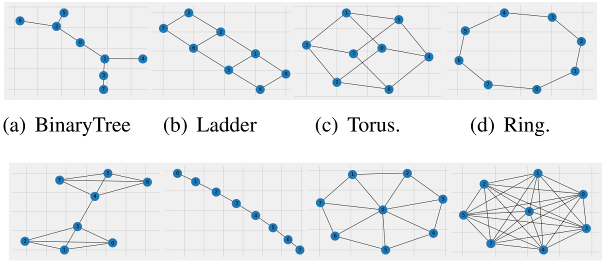
[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016.

[8] G. Neglia, C. Xu, D. Towsley, and G. Calbi, "Decentralized gradient methods: does topology matter?" in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2348–2358.

Appendix

A. Appendix A



(a) BinaryTree (b) Ladder (c) Torus. (d) Ring.

(e) Barbell (f) Chain (g) Wheel (h) Fully connected.

Fig. 5. Different choices of topolgies. Note we only conduct experiments with 8 nodes in our project, so 3 dimensional torus topology is like hyper-cube.

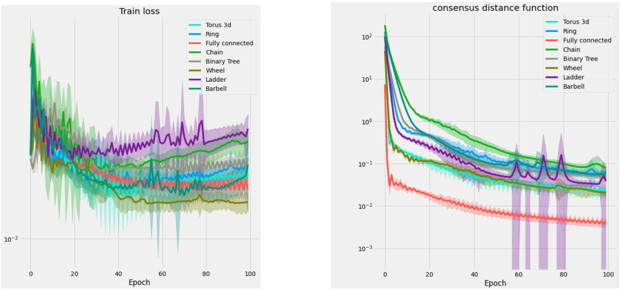
B. Appendix B

```
Algorithm 2 DECENTRALIZED SGD
Input: $X^{(0)}$, stepsizes $\{\eta_t\}_{t=0}^{T-1}$, number of iterations $T$, mixing matrix distributions $\mathcal{W}^h(t), t \in 0, ..., T$
1: for $t$ in $0, ..., T$ do in parallel on all workers
2:     $G_t = \partial F(X^{(t)}, \xi^{(t)})$                ▷ stochastic gradients
3:     $W^{(t)} \sim \mathcal{W}^{(t)}$                      ▷ sample mixing matrix
4:     $X^{t+1} = (X^t - \eta_t G^{(t)} W^{(t)})$            ▷ update & mixing
   End For Parallel
```

On line 2 of Algorithm 2 every node in parallel calculates stochastic gradients, on line 3 a mixing matrix is sampled from the distribution $\mathcal{W}$, and on line 4, every node performs a local SGD update and after that mixes updated parameters with the sampled matrix $W^{(t)}$.

C. Appendix C

We also conducted experiments on D-SGD with different topologies and one label skewed non-iid data. The results is presented in Fig. 6. It is obvious that the convergence curses are affected by this type of non-iid setting. The effect is similar to that of two label skewed non-iid setting.



(a) Train loss (b) Consensus distance function

Fig. 6. Convergence of D-SGD with different topologies on one label skewed non-iid data.

D. Appendix D

In here we add some interesting details behind the main plots in the report.

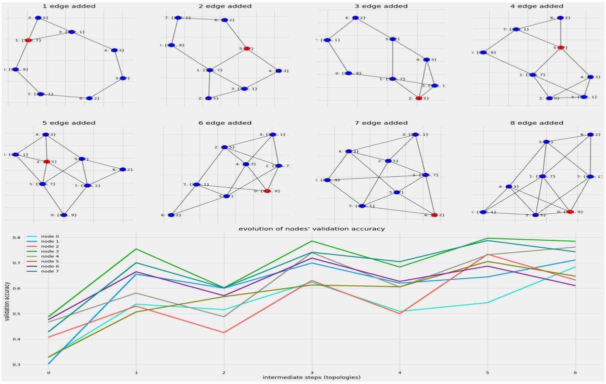


Fig. 7. Algorithm 1's steps on given distribution and its node validation accuracy

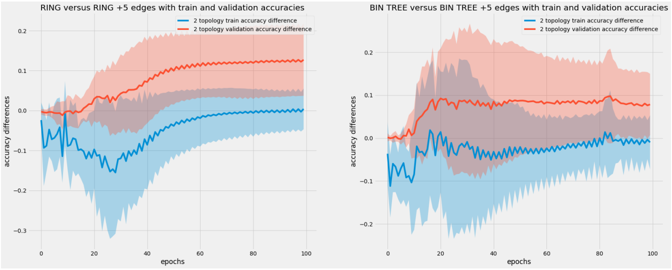


Fig. 8. Difference in accuracy between original and modified topology for various starting topology