ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447268

FRIENDS TO HELP: SAVING FEDERATED LEARNING FROM CLIENT DROPOUT

Heqiang Wang, Jie Xu

Electrical and Computer Engineering, University of Miami
Coral Gables, FL 33146, USA

ABSTRACT

Federated learning (FL) is a new distributed machine learning framework known for its benefits on data privacy and communication efficiency. Since full client participation in many cases is infeasible due to constrained resources, partial participation FL algorithms have been investigated that *proactively* select/sample a subset of clients, aiming to achieve learning performance close to the full participation case. This paper studies a *passive* partial client participation scenario that is much less well understood, where partial participation is a result of external events, namely client dropout, rather than a decision of the FL algorithm. We cast FL with client dropout as a special case of a larger class of FL problems where clients can submit substitute (possibly inaccurate) local model updates. Based on our convergence analysis, we develop a new algorithm FL-FDMS that discovers friends of clients (i.e., clients whose data distributions are similar) on-the-fly and uses friends' local updates as substitutes for the dropout clients, thereby reducing the substitution error. Experiments on MNIST and CIFAR-10 confirmed the superior performance of FL-FDMS in handling client dropout in FL.

Index Terms— Federated learning, client dropout, bias mitigation.

1. INTRODUCTION

Federated learning (FL) is a distributed machine learning paradigm where a set of clients with decentralized data work collaboratively to learn a model under the coordination of a centralized server. Depending on whether or not all clients participate in every learning round, FL is classified as either *full participation* or *partial participation*. While full participation is the ideal FL mode that achieves the best convergence performance, a lot of effort has been devoted to developing partial participation strategies via client selection/sampling [1–9] due to the attractive benefit of reduced resource (i.e. communication and computation) consumption. Existing works show that some of these partial participation strategies [2, 3] can indeed achieve performance close to full participation. Although the details differ, the principal idea of these strategies is the careful selection of *appropriate* clients to participate in each FL round. For example, in many cases [1–3], clients are sampled uniformly at random so that the participating clients form an "unbiased" representation of the whole client population in terms of the data distribution. In others [4–9], "important" clients are selected more often to lead FL towards the correct loss descending direction.

This paper studies partial participation FL, but from an angle in stark contrast with existing works. In our considered problem, partial participation is a result of an arbitrary client dropout process, which the FL algorithm has absolutely no control over. However, a client may not be able to participate (in other words, drop out) in an FL round due to, e.g., dead/low battery or loss of the communication signal. This means that the subset of clients participating in a FL round may not be "representative" or "important" in any sense. Client dropout is related to the "straggler" issue in FL, which is caused by the delayed local model uploading by some clients. Existing solutions to the straggler issue can be categorized into the following two types: allowing clients to upload their local models asynchronously to the server [10–13], and using the stored last updates of the inactive clients to join the model aggregation [14, 15].

We shall note that client dropout can occur simultaneously with client selection/sampling and hence partial participation can be a mixed result of both. As will become clear, our algorithm can be readily applied to this scenario and our theoretical results can also be extended provided that the client selection/sampling strategy used in conjunction has its own theoretical performance guarantee. However, since these results will depend on the specific client selection/sampling strategy adopted, and in order to better elucidate our main idea, this paper will not consider client selection/sampling. Our main contributions are summarized as follows: (1) We analyze FL problems with inaccurate local updates, including client dropout, and find that FL convergence depends on the gap between actual and substitute updates. Minimizing this gap is key for better FL performance with dropout. (2) We introduce "friendship" among clients with similar data and updates. To mitigate dropout effects, we use a non-dropped friend's update, but identifying these friendships is challenging. (3) Our method dynamically identifies friendships for update substitution. Tests on MNIST and CIFAR-10 show their effectiveness in improving FL performance with client dropout.

2. FEDERATED LEARNING WITH CLIENT DROPOUT

We consider a server and a set of K clients, who work together to train a machine learning model by solving a distributed optimization problem:

$$\min_{w\in\mathbb{R}^d}\left\{f(w):=\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{\xi^k\sim\mathcal{D}^k}[F^k(w;\xi^k)]\right\} \tag{1}$$

where $F^k:\mathbb{R}^d\to\mathbb{R}$ denotes the objective function, $\xi^k\sim\mathcal{D}^k$ represents the sample/s drawn from distribution $\mathcal{D}^k$ at the k-th client and $w\in\mathbb{R}^d$ is the model parameter to learn. In a non-i.i.d. data setting, the distributions $\mathcal{D}^k$ are different across the clients.

We consider a typical FL algorithm [16] working in the client dropout setting. In each round t, only a subset $\mathcal{S}_t\subseteq\mathcal{K}$ of clients participate due to external reasons uncontrollable by the FL algorithm. We call the clients that cannot participate *dropout* (or *inactive*) clients. Then, FL executes the following four steps among the *non-dropout* (or *active*) clients in round t:

This work is supported in part by the National Science Foundation under grants 2033681, 2006630, 2044991, 2319780.

1. Global model download. Each client $k \in \mathcal{S}_t$ downloads the global model w_t from the server.

2. Local model update. Each client $k \in \mathcal{S}_t$ uses w_t as the initial model to train a new local model w_{t+1}^k, typically by using mini-batch stochastic gradient descent (SGD) as follows:

$$w_{t,\tau+1}^k = w_{t,\tau}^k - \eta_L g_{t,\tau}^k, \forall \tau = 1, ..., E \tag{2}$$

where $\xi_{t,\tau}^k$ is a mini-batch of data samples, $g_{t,\tau}^k = \nabla F^k(w_{t,\tau}^k; \xi_{t,\tau}^k)$ is the mini-batch stochastic gradient, η_L is the client local learning rate and E is the number of epochs for local training.

3. Local model upload. Clients upload their local model updates to the server. Instead of uploading the local model w_{t+1}^k itself, client k can simply upload the *local model update* Δ_t^k, which is defined as the accumulative model parameter difference as follows:

$$\Delta_t^k = \frac{1}{\eta_L}\left(w_{t,E}^k - w_{t,0}^k\right) = -\sum_{\tau=0}^{E-1} g_{t,\tau}^k \tag{3}$$

4. Global model update. The server updates the global model by using the aggregated local model updates of the clients in $\mathcal{S}_t$:

$$w_{t+1} = w_t + \eta\eta_L \Delta_t, \quad \text{where} \quad \Delta_t := \frac{1}{S_t}\sum_{k\in\mathcal{S}_t} \Delta_t^k \tag{4}$$

and η is the global learning rate and $S_t \triangleq |\mathcal{S}_t|$ denotes the number of the non-dropout clients.

For the main result of this paper, we consider the most general case of the client dropout process by imposing only an upper limit on the dropout ratio. That is, there exists a constant $\alpha \in [0, 1)$ such that $(K - S_t)/K \leq \alpha$. If all clients drop out in a round, then essentially the round is skipped. Also note that if $\mathcal{S}_t$ were a choice of the FL algorithm, then the problem would become FL with client selection/sampling. We stress again that in our problem, $\mathcal{S}_t$ is not a choice, it is an uncontrollable client participation scenario.

3. CONVERGENCE ANALYSIS

Consider an FL round t where the set $\mathcal{S}_t$ of clients are active while the remaining set $\mathcal{K}\backslash\mathcal{S}_t$ of clients dropped out. Thus, one can only use the local model updates Δ_t^k of the active clients in $\mathcal{S}_t$ to perform global model updates since the inactive clients upload nothing to the server. However, rather than completely ignoring the inactive clients, we write the aggregate model update Δ_t in a different way to include all clients in the equation:

$$\Delta_t := \frac{1}{S_t}\sum_{k\in\mathcal{S}_t} \Delta_t^k = \frac{1}{K}\left(\sum_{k\in\mathcal{S}_t} \Delta_t^k + \sum_{k\in\mathcal{K}\backslash\mathcal{S}_t} \tilde{\Delta}_t^k\right) \tag{5}$$

where in the second equality we simply take $\tilde{\Delta}_t^k = \frac{1}{S_t}\sum_{k\in\mathcal{S}_t} \Delta_t^k$. In other words, although the inactive clients did not participate in the round t's learning, it is equivalent to the case where an inactive client $k \in \mathcal{K}\backslash\mathcal{S}_t$ uses $\tilde{\Delta}_t^k = \frac{1}{S_t}\sum_{k\in\mathcal{S}_t} \Delta_t^k$ as a substitute of its true local update Δ_t^k (which it may not even calculate due to dropout). Apparently, because $\tilde{\Delta}_t^k \neq \Delta_t^k$ in general, similar substitutes lead to a biased error in the global update and hence affect the FL convergence performance.

Leveraging the above observation, we consider a larger class of FL problems that include client dropout as a special case. Specifically, imagine that an inactive client k, instead of contributing nothing, uses a substitute $\tilde{\Delta}_t^k$ for Δ_t^k when submitting its local model update. Apparently, $\tilde{\Delta}_t^k = \frac{1}{S_t}\sum_{k\in\mathcal{S}_t} \Delta_t^k$ is a specific choice of the substitute. We will still use the notation Δ_t as the aggregate model update with local update substitution and the readers should not be confused. Our convergence analysis will utilize the following standard assumptions about the FL problem.

Assumption 1 (Lipschitz Smoothness)**.** *The local objective functions satisfy the Lipschitz smoothness property, i.e.,*$\exists L > 0$*, such that* $\|\nabla F^k(x) - \nabla F^k(y)\| \leq L\|x - y\|$*,* $\forall x, y \in \mathbb{R}^d$ *and* $\forall k \in \mathcal{K}$*.*

Assumption 2 (Unbiased Local Gradient Estimator)**.** *The mini-batch based local gradient estimator is unbiased, i.e.*
$\mathbb{E}_{\xi^k\sim\mathcal{D}^k}[\nabla F^k(x; \xi^k)] = \nabla F^k(x), \forall k \in \mathcal{K}$

Assumption 3 (Bounded Local and Global Variance)**.** *There exist constants* $\rho_L > 0$ *and* $\rho_G > 0$ *such that the variance of each local gradient estimator is bounded, i.e.,*
$\mathbb{E}_{\xi^k\sim\mathcal{D}^k}\left[\|\nabla F^k(x; \xi^k) - \nabla F^k(x)\|^2\right] \leq \rho_L^2, \forall x, \forall k \in \mathcal{K}$
And the global variability of the local gradient is bounded by $\|\nabla F^k(x) - \nabla f(x)\|^2 \leq \rho_G^2, \forall x, \forall k \in \mathcal{K}$*.*

Let $\bar{\Delta}_t = \frac{1}{K}\sum_{k\in\mathcal{K}} \Delta_t^k$ be the average local model update assuming that all clients are active in round t and submitted their true local updates. Thus, $e_t := \Delta_t - \bar{\Delta}_t$ represents aggregate global update error due to client dropout and local update substitution in round t. Furthermore, let $e_t^k := \tilde{\Delta}_t^k - \Delta_t^k$, $\forall k \in \mathcal{K}\backslash\mathcal{S}_t$ be the individual substitution error for an individual inactive client k in round t.

Theorem 1. *Let constant local and global learning rates* η_L *and* η *be chosen as such that* $\eta_L \leq \frac{1}{8EL}$ *and* $\eta\eta_L \leq \frac{1}{4EL}$*. Under Assumption 1-3, the sequence of model* w_t *generated by using model update substitution with a substitution error sequence* $e_0, ..., e_{T-1}$ *satisfies*

$$\min_{t=0,...,T-1} \mathbb{E}\|\nabla f(w_t)\|^2 \leq \frac{f_0 - f_*}{c\eta\eta_L ET} + \Phi + \Psi(e_0, ..., e_{T-1}) \tag{6}$$

where $\Phi = \frac{1}{c}\left[5\eta_L^2 EL^2(\rho_L^2 + 6E\rho_G^2) + \frac{\eta\eta_L L}{K}\rho_L^2\right]$*,* c *is a constant,* $f_0 \triangleq f(w_0)$*,* $f_* \triangleq f(w_*)$*,* w_* *is the optimal model and*

$$\Psi(e_0, ..., e_{T-1}) = \frac{1 + 3\eta\eta_L LE}{cE^2T}\sum_{t=0}^{T-1} \mathbb{E}[\|e_t\|^2] \tag{7}$$

The expectation is over the local dataset samples among the clients.

The convergence bound consists of three components: a diminishing term as T grows, a constant term Φ independent of T, and a term based on substitution errors $e_0, ..., e_{T-1}$. With constant learning rates η and η_L and bounded $\|e_t\|^2$, the convergence bound is $O(1/T) + C$, where C is a constant. The key insight derived by Theorem 1 is that the FL convergence bound depends on the cumulative substitution error $\sum_{t=0}^{T-1} \mathbb{E}_t[\|e_t\|^2]$. Without client dropout, this error is zero, making the convergence bound simply $\frac{f_0 - f_*}{c\eta\eta_L ET} + \Phi$, which degenerates to the same bound established in [3] for the normal full participation case.

Our analysis provides an intuitive understanding of the impact of client dropout and model substitution on FL convergence and characterizes the convergence under a biased scenario, which is an important and previously unaddressed issue in FL research. The main challenge was consolidating all the bias-related errors into a single term in the final convergence bound.

Next, we derive a more specific bound on $\Psi(e_0, ..., e_{T-1})$ for the naive dropout case where an inactive client uploads nothing to the server or, equivalently, uses $\frac{1}{S_t}\sum_{k\in\mathcal{S}_t}\Delta_t^k$ as a substitute. The following additional assumption is needed.

Assumption 4. *For any two clients i and j, the local model update difference is bounded as follows: $\mathbb{E}[\|\Delta^i(w)-\Delta^j(w)\|^2] \leq \sigma_{i,j}^2, \forall w$ where the expectation is over the local dataset samples.*

Assumption 4 provides a pairwise characterization of clients' dataset heterogeneity in terms of the local model updates. When two clients i, j have the same data distribution and assuming that the min-batch SGD utilizes the entire local dataset (i.e., the local gradient estimator is accurate), then it is obvious $\sigma_{i,j}^2 = 0$. We let $\sigma_P^2 \triangleq \max_{i,j}\sigma_{i,j}^2$ be the maximum pairwise difference.

With Assumption 4, the round-t substitution error can then be bounded as $\mathbb{E}[\|e_t\|^2] \leq \alpha^2\sigma_P^2$. Plugging this bound into $\Psi(e_0, ..., e_{T-t})$, we have

$$\Psi(e_0, ..., e_{T-1}) \leq \frac{\alpha^2\sigma_P^2(1+3\eta\eta_L LE)}{cE^2} \triangleq \bar{\Psi} \tag{8}$$

Note that $\bar{\Psi}$ is a constant independent of T. This implies that, with constant learning rates η_L and η, $\min_t \mathbb{E}\|\nabla f(w_t)\|^2$ converges to some value at most $\Phi + \bar{\Psi}$ as $T \to \infty$.

4. FRIEND MODEL SUBSTITUTION

In this section, we develop a new algorithm to reduce the substitution error of FL with client dropout. Our key idea is to find a better substitute $\tilde{\Delta}_t^k$ for Δ_t^k when client k drops out in round t in order to reduce $\Psi(e_0, ..., e_{T-1})$. This is possible by noticing that $\sigma_{i,j}^2$ are different across client pairs and the local model updates are more similar when the clients' data distributions are more similar. Thus, when a client i drops out, one can use the local model update Δ_t^j as a replacement of Δ_t^i if j shares a similar data distribution with i, or in our terminology, j is a friend of i. We make "friendship" formal in the following definition.

Definition 1 (Friendship)**.** *Let $\sigma_F^2 < \sigma_P^2$ be some constant. We say that clients i and j are friends if $\sigma_{i,j}^2 \leq \sigma_F^2$. Further, denote $\mathcal{B}_k$ as the set of friends of client k and $B_k = |\mathcal{B}_k|$ as the size of $\mathcal{B}_k$.*

While "friendship" exists among clients, it's **hidden** in the FL algorithm, making model update substitution for dropout clients challenging. However, knowing this, if friendship was fully revealed, would provide an optimal baseline for our following proposed algorithm.

Suppose in any round t, for any inactive client $i \in \mathcal{K}\backslash\mathcal{S}_t$, there exists active client j that is client i's friend. With friend model substitution, the accumulated substitution error can be bounded as $\mathbb{E}[\|e_t\|^2] \leq \alpha^2\sigma_F^2$. Substituting this bound into $\Psi(e_0, ..., e_{T-1})$:

$$\Psi(e_0, ..., e_{T-1}) \leq \frac{\alpha^2\sigma_F^2(1+3\eta\eta_L LE)}{cE^2} \triangleq \Psi^* \tag{9}$$

Note that Ψ^* is still a constant independent of T but it is much smaller than $\bar{\Psi}$, since typically $\sigma_F^2 \ll \sigma_P^2$. In Fig. 1, we illustrate the difference in local model updates between σ_F^2 and σ_P^2 throughout the entire training period. From the result, we observe that σ_F^2 is much smaller than σ_P^2 in every round. Therefore, the convergence bound can be improved if the algorithm utilizes the friendship information.

The above analysis shows that the FL convergence can be substantially improved if the algorithm can utilize the friendship information. Next, we develop a learning-assisted FL algorithm, called

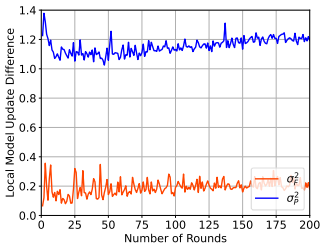


Fig. 1: The Local Model Update Difference σ_F^2 and σ_P^2.

FL with Friend Discovery and Model Substitution (FL-FDMS), that discovers the friends of clients and uses the model update of the discovered friends for substitution.

In each FL round t, in addition to the regular steps in the FL algorithm described in Section 3, FL-FDMS performs different actions depending on the client status. For active clients, FL-FDMS calculates pairwise similarity scores to learn the similarity between clients. For inactive clients, FL-FDMS uses the historical similarity scores to find active friend clients and use their local model updates as substitutes. We describe these two cases in more detail below.

Active Clients. For any pair of active clients i and j in $\mathcal{S}_t$. The server calculates a similarity score $r_t^{i,j} = r(\Delta_t^i, \Delta_t^j)$ based on their uploaded local model updates Δ_t^i and Δ_t^j. Many functions can be used to calculate the score. For example, $r(\Delta_t^i, \Delta_t^j)$ can simply be the negative model difference, i.e., $-\|\Delta_t^i - \Delta_t^j\|$, or the normalized cosine similarity, i.e.,

$$r(\Delta_t^i, \Delta_t^j) = \frac{1}{2}\left(\frac{\langle\Delta_t^i, \Delta_t^j\rangle}{\|\Delta_t^i\|\|\Delta_t^j\|} + 1\right) \tag{10}$$

Because the normalized cosine similarity takes value from a bounded and normalized range $[0, 1]$, which is more amenable for mathematical analysis, we will use this function in this paper. Clearly, a higher similarity score implies that the two clients are more similar in terms of their data distribution.

However, a single similarity score calculated in one particular round does not provide accurate similarity information because of the randomness in the initial model in that round and the randomness in the mini-batch SGD for local model computation. Thus, the server maintains and updates an average similarity score $R_t^{i,j}$ for clients i and j based on all similarity scores calculated so far as follows,

$$R_t^{i,j} = \begin{cases} \frac{N_{t-1}^{i,j}}{N_{t-1}^{i,j}+1}R_{t-1}^{i,j} + \frac{1}{N_{t-1}^{i,j}+1}r_t^{i,j}, & \text{if } i,j \in \mathcal{S}_t \\ R_{t-1}^{i,j}, & \text{otherwise} \end{cases} \tag{11}$$

where $N_t^{i,j}$ is the number of rounds where both clients i and j did not drop out up to round t.

Inactive Clients. For any inactive client k, the server looks up $R_t^{k,i}$ between k and every active client $i \in \mathcal{S}_t$, finds the one with the highest similarity score, denoted by $\phi_t(k) = \arg\max_{i\in\mathcal{S}_t} R_t^{k,i}$, and uses the local model update $\Delta_t^{\phi_t(k)}$ as a substitute for Δ_t^k when computing the global update.

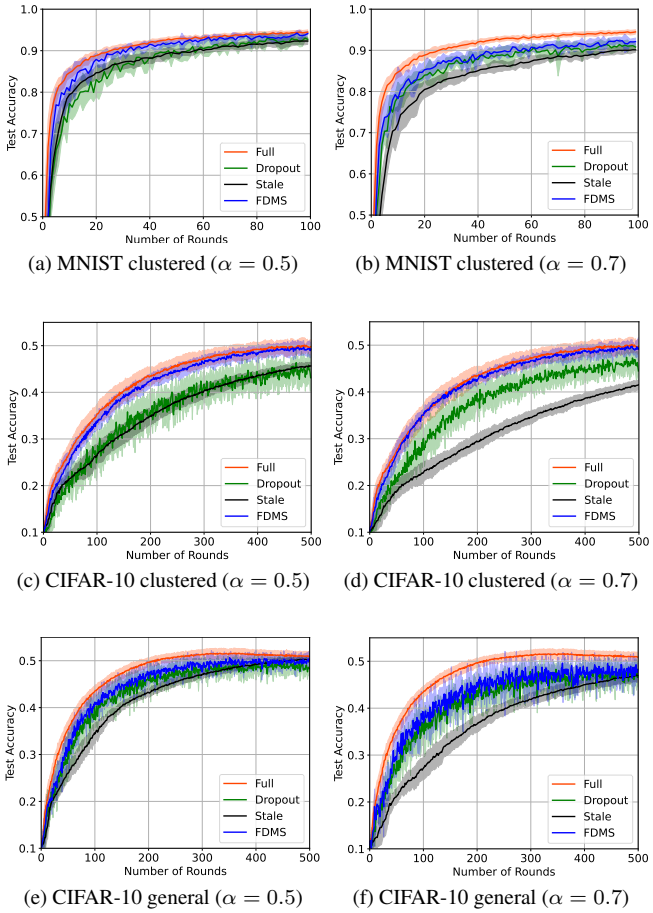
In the experimental section, we will demonstrate that our FL-FDMS algorithm effectively identifies hidden friends for clients who drop out, leading to improved learning performance.

Remark on Privacy: The averaged similarity score is calculated based on the uploaded model and does not require any additional information from the client. Previous works such as [17, 18], albeit addressing different FL problems, also rely on finding the client relationships or clusters. Thus, the privacy protection level of our algorithm is similar to that of those algorithms.

5. EXPERIMENTS

5.1. Setup

We perform experiments on two standard public datasets, namely MNIST and CIFAR-10, with two data settings. In the clustered settings (one on MNIST and one on CIFAR-10), we artificially create 5 client clusters where clients in the same cluster possess data samples with the same labels. Thus, clients in the same cluster are naturally regarded as friends. However, the clustering structure is *unknown* to our algorithm. In the general setting (on CIFAR-10), 20 clients receive a random subset of the whole dataset using a common way of generating non-iid FL datasets that are widely used in existing works. We use the LeNet architecture [19] to train the MNIST and CIFAR-10 datasets. The following parameters are used for training: the number of local iterations $E = 2$, the local learning rate $\eta_L = 0.1$ and the global learning rate $\eta = 1$.



(a) MNIST clustered ($\alpha = 0.5$) (b) MNIST clustered ($\alpha = 0.7$)

(c) CIFAR-10 clustered ($\alpha = 0.5$) (d) CIFAR-10 clustered ($\alpha = 0.7$)

(e) CIFAR-10 general ($\alpha = 0.5$) (f) CIFAR-10 general ($\alpha = 0.7$)

Fig. 2: Performance comparison with various α

We compare FL-FDMS with the following three benchmarks.

Full Participation (Full). This is the ideal case where all clients participate in FL without dropout. It is used as a performance upper bound.

Client Dropout (Dropout). In this case, the server simply ignores the dropout clients and performs global aggregation on the non-dropout clients.

Staled Substitute (Stale). Another method to deal with dropout clients is to use their last uploaded local model updates for the current round's global aggregation. Such a method was also used to deal with the "straggler" issue in FL in some previous works [14, 15].

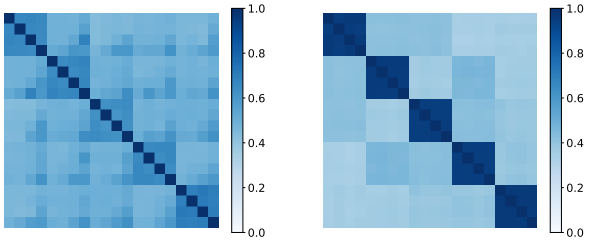
5.2. Performance Comparison

We first compare the convergence performance in the clustered setting under different dropout ratios $\alpha \in \{0.5, 0.7\}$. Fig. 2 plots the convergence curves on the MNIST dataset and the CIFAR-10 dataset, respectively. Several observations are made as follows. First, **FL-FDMS** outperforms **Dropout** and **Stale** in terms of test accuracy and convergence speed and achieves performance close to **Full** in all cases. Second, **FL-FDMS** reduces the fluctuations caused by the client dropout on the convergence curve. Third, with a larger dropout ratio, the performance improvement of **FL-FDMS** is larger. Fourth, on more complex datasets (e.g., CIFAR-10), **FL-FDMS** achieves an even more significant performance improvement.

We also perform experiments in the more general non-iid case to illustrate the wide applicability of the proposed algorithm. Fig. 2 plots the convergence curves on CIFAR-10 under the general setting. The results confirm the superiority of **FL-FDMS**. However, we also note that the improvement is smaller than that in the clustered setting. This suggests a limitation of **FL-FDMS**, which works best when the "friendship" relationship among the clients is stronger.

5.3. Friend Discovery

FL-FDMS relies on successfully discovering the friends of dropout clients. In Fig. 3, we show the pairwise similarity scores in the final learning round. In our controlled clustered setting, 20 clients were grouped into 5 clusters, but this information was not known by the algorithm at the beginning. As the figure shows, the similarity scores obtained by **FL-FDMS** are larger for intra-cluster client pairs and smaller for inter-cluster client pairs, indicating that the clustering/friendship information can be successfully discovered. Moreover, our experiments show that the discovered friendship is more obvious for CIFAR-10 than for MNIST. This is likely due to the different dataset structures and the different CNN models adopted.



(a) MNIST dataset (b) CIFAR-10 dataset

Fig. 3: Pairwise similarity scores in the final learning round.

6. CONCLUSION

This paper investigated the impact of client dropout on the convergence of FL. Our analysis treats client dropout as a special case of local update substitution and characterizes the convergence bound in terms of the total substitution error. This inspired us to develop FL-FDMS, which discovers friend clients on-the-fly and uses friends' updates to reduce substitution errors, thereby mitigating the negative impact of client dropout. Extensive experiment results show that discovering the client's "friendship" is possible and it can be a useful resort for addressing client dropout problems.

7. REFERENCES

[1] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning.," 2019.

[2] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2019.

[3] Haibo Yang, Minghong Fang, and Jia Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," in *International Conference on Learning Representations*, 2020.

[4] Monica Ribero and Haris Vikalo, "Communication-efficient federated learning via optimal client sampling," *arXiv preprint arXiv:2007.15197*, 2020.

[5] Wenlin Chen, Samuel Horváth, and Peter Richtárik, "Optimal client sampling for federated learning," *Transactions on Machine Learning Research*, 2022.

[6] Yae Jee Cho, Jianyu Wang, and Gauri Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[7] Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *15th Symposium on Operating Systems Design and Implementation*, 2021, pp. 19–35.

[8] Hongda Wu and Ping Wang, "Node selection toward faster convergence for federated learning on non-iid data," *IEEE Transactions on Network Science and Engineering*, 2022.

[9] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes, "Diverse client selection for federated learning via submodular maximization," in *International Conference on Learning Representations*, 2021.

[10] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, and Stephen Jarvis, "Safa: a semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2020.

[11] Yanan Li, Shusen Yang, Xuebin Ren, and Cong Zhao, "Asynchronous federated learning with differential privacy for edge intelligence," *arXiv preprint arXiv:1912.07902*, 2019.

[12] Cong Xie, Sanmi Koyejo, and Indranil Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[13] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 15–24.

[14] Yikai Yan, Chaoyue Niu, Yucheng Ding, Zhenzhe Zheng, Fan Wu, Guihai Chen, Shaojie Tang, and Zhihua Wu, "Distributed non-convex optimization with sublinear speedup under intermittent client availability," *arXiv preprint arXiv:2002.07399*, 2020.

[15] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang, "Fast federated learning in the presence of arbitrary device unavailability," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12052–12064, 2021.

[16] Jakub Konevcny, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[17] Yichen Ruan and Carlee Joe-Wong, "Fedsoft: Soft clustered federated learning with proximal local updating," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 8124–8131.

[18] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19586–19597, 2020.

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.