Multi-Center Federated Learning

Ming Xie[1] , **Guodong Long**[1] , **Tao Shen**[1] , **Tianyi Zhou**[2] ,
Xianzhi Wang[1] , **Jing Jiang**[1] , **Chengqi Zhang**[1]
[1]Australian AI Institute, Faculty of Engineering and IT, University of Technology Sydney
[2]Paul G. Allen School of Computer Science and Engineering, University of Washington
ming.xie@student.uts.edu.au, {guodong.long, tao.shen, xianzhi.wang, jing.jiang}@uts.edu.au,
tianyizh@uw.edu

Abstract

Federated learning (FL) can protect data privacy in distributed learning since it merely collects local gradients from users without access to their data. However, FL is fragile in the presence of heterogeneity that is commonly encountered in practical settings, e.g., non-IID data over different users. Existing FL approaches usually update a single global model to capture the shared knowledge of all users by aggregating their gradients, regardless of the discrepancy between their data distributions. By comparison, a mixture of multiple global models could capture the heterogeneity across various users if assigning the users to different global models (i.e., centers) in FL. To this end, we propose a novel multi-center aggregation mechanism . It learns multiple global models from data, and simultaneously derives the optimal matching between users and centers. We then formulate it as a bi-level optimization problem that can be efficiently solved by a stochastic expectation maximization (EM) algorithm. Experiments on multiple benchmark datasets of FL show that our method outperforms several popular FL competitors.

1 Introduction

The widespread of mobile phones and Internet-of-Things has witnessed a huge volume of data generated by end-users on mobile devices. Generally, a service provider on the server side collect users' data and train a global machine learning model such as deep neural networks. Such a centralized machine learning approach causes severe practical issues, e.g., communication costs, consumption of device batteries, and the risk of violating the privacy and of user data.

Federated learning (FL) [McMahan *et al.*, 2017] is a decentralized machine learning framework that learns models collaboratively using the training data distributed on remote devices to boost communication efficiency. Basically, it learns a shared pre-trained model by aggregating the locally-computed updates, and each update is derived from learning the data in the corresponding local device. Therefore, a straightforward aggregation algorithm is responsible for averaging the many local models' parameters, weighted by the size of the training data on each device. Compared with conventional distributed machine learning, FL is robust against unbalanced and non-IID data distributions, which is the defining characteristic of modern AI products for mobile devices.

The vanilla FL addresses a practical setting of distributed learning, where 1) the central server is not allowed to access any user data which protects users' privacy, and 2) the data distribution over different users is non-IID, which is a natural assumption of real-world applications. However, early FL approaches [McMahan *et al.*, 2017; Yang *et al.*, 2019] use only one global model as a single-center to aggregate the information of all users. The stochastic gradient descent (SGD) for single-center aggregation is designed for IID data, and therefore, conflicts with the non-IID setting in FL.

Recently, the non-IID or heterogeneity challenge of FL has been studied to improve the robustness of global models against outlier/adversarial users and devices [Ghosh *et al.*, 2019; Li *et al.*, 2018; Li *et al.*, 2019c]. Moreover, Sattler *et al.* [2019] proposed an idea of clustered FL (FedCluster) that addresses the non-IID issue by dividing the users into multiple clusters. However, the hierarchical clustering in FedCluster is achieved by multiple rounds of bipartite separation, each requiring the federated SGD algorithm to run until convergence. Hence, its computational and communication efficiency will become bottlenecks when applied to a large-scale FL system. More recently, Mansour *et al.* [2020] and Ghosh *et al.* [2020] proposed to cluster the local models according to the loss of hypothesis. In particular, each user will try all K global models representing K clusters, and then select the best global model as the cluster ID by considering the lowest loss of running the global model on local data. However, this posts high communication and computation overheads because the selected nodes will spend more resources for receiving and running multiple global models.

In this paper, we propose a novel multi-center FL framework that updates multiple global models by aggregating information from multiple user groups. In particular, the datasets of the users in the same group are likely to be generated or derived from the same or similar distribution. We formulate the problem of the multi-center FL as the joint clustering of users, and then optimizing of the global model for users in each cluster. In particular, (1) each user's local model is assigned to its closest global model, and (2) the global model in each cluster leads to the smallest loss over all the associated users. The proposed multi-center FL not only inherits the communication efficiency of the federated SGD but also retains the capability of handling non-IID data on heterogeneous datasets. Lastly, we propose a new optimization method in line with EM algorithm to train our model.

We summarise our main contributions as:

- We propose a novel multi-center aggregation approach (Section 4.1) to address the non-IID challenge of FL.
- We design an objective function, namely multi-center federated loss (Section 4.2), for user clustering in FL.
- We propose Federated Stochastic Expectation Maximization (FeSEM) (Section 4.3) to solve the optimization of the proposed objective function.
- We present the algorithm as an easy-to-implement and strong baseline for FL. Its effectiveness is evaluated on benchmark datasets. (Section 6)

2 Related work

Federated learning (FL) enables users to leverage rich data machine learning models without compromising their data. It has attracted a significant amount of research interest since 2017, with many studies investigating FL from several aspects, e.g., system perspective, personalized models, scalability [Bonawitz *et al.*, 2019], communication efficiency [Konecný *et al.*, 2018], and privacy [Geyer *et al.*, 2017]. Most of the related work addresses a particular concern such as security or privacy [Rouhani *et al.*, 2018; Liu *et al.*, 2019; Cao *et al.*, 2020].

FL is designed for specific scenarios that can be further expanded to a standard framework to preserve data privacy in large-scale machine learning systems or mobile edge networks [Lim *et al.*, 2020]. For example, [Yang *et al.*, 2019] expanded FL by introducing a comprehensive, secure FL framework that includes horizontal FL, vertical FL, and federated transfer learning. The work in [Li *et al.*, 2019b; Lyu *et al.*, 2020] surveyed the FL systems in relation to their functions on privacy protection and security threats. [Kairouz *et al.*, 2019] discussed the advances and open problems in FL. [Caldas *et al.*, 2018] proposed LEAF – a benchmark for federated settings with multiple datasets. [Luo *et al.*, 2019] proposed an object detection-based dataset for FL.

Heterogeneity is a core challenge in the federated setting and has been widely studied from various perspectives. [Haddadpour and Mahdavi, 2019] conducted theoretical convergence analysis for FL with heterogeneous data. [Hsu *et al.*, 2019] measured the effects of non-IID data for federated visual classification. [Yang *et al.*, 2020] proposed a heterogeneity-aware platform design for FL. [Liang *et al.*, 2020] discussed the local representations that enable data to be processed on new devices in different ways according to their source modalities instead of using a single global model. The single global model might not generalize to unseen modalities and distributions of data. [Li and Wang, 2019] proposed a new federated setting composed of a shared global dataset and many heterogeneous datasets from devices. [Jeong *et al.*, 2018] and [Lin *et al.*, 2020] proposed to integrate knowledge distillation with FL to tackle the model heterogeneity. [Yu *et al.*, 2020] proposed a general FL framework to align heterogeneous model architectures and functional neurons.

To solve the problem caused by non-IID data in a federated setting [Caldas *et al.*, 2018], [Sattler *et al.*, 2019] proposed clustered FL (FedCluster) by integrating FL and bi-partitioning-based clustering into an overall framework, and [Mansour *et al.*, 2020; Ghosh *et al.*, 2020] proposed a hypothesis-based federated clustering that assigns the cluster by considering the loss of running the global model on local data. [Ghosh *et al.*, 2019] proposed a robust FL comprising three steps: 1) learning a local model on each device, 2) clustering model parameters to multiple groups, each being a homogeneous dataset, and 3) running a robust distributed optimization [Li *et al.*, 2019a] in each cluster.

[Li *et al.*, 2019c] proposed FedDANE by adapting the DANE [Shamir *et al.*, 2014] to a federated setting. In particular, FedDANE is a federated Newton-type optimization method. [Li *et al.*, 2018] proposed FedProx for the generalization and re-parameterization of FedAvg [McMahan *et al.*, 2017]. It adds a proximal term to the objective function of each device's supervised learning task, and the proximal term is to measure the parameter-based distance between the server and the local model. [Arivazhagan *et al.*, 2019] added a personalized layer for each local model, i.e., FedPer, to tackle heterogeneous data.

3 Background

3.1 Problem Setting

In FL, each device-i has a private dataset $\mathcal{D}_i = \{\mathcal{X}_i, \mathcal{Y}_i\}$, where $\mathcal{X}_i$ and $\mathcal{Y}_i$ denote the input features and corresponding gold labels respectively. Each dataset $\mathcal{D}_i$ will be used to train a local supervised learning model $\mathcal{M}_i : \mathcal{X}_i \rightarrow \mathcal{Y}_i$. $\mathcal{M}$ denotes a deep neural model parameterized by weights W. It is built to solve a specific task, and all devices share the same model architecture.

For the i-th device, given a private training set $\mathcal{D}_i$, the training procedure of $\mathcal{M}_i$ is represented in brief as

$$\min_{W_i} L_s(\mathcal{M}_i, \mathcal{D}_i, W_i), \tag{1}$$

where $L_s(\cdot)$ is a general definition of the loss function for any supervised learning task, and its arguments are model structure, training data and learnable parameters respectively, and W' denotes the parameters after training. In general, the data from one device is insufficient to train a data-driven neural network with satisfactory performance. An FL framework optimizes the local models in a distributed manner and minimizes the loss of the local data on each device.

Hence, the optimization in vanilla FL over all the local models can be written as

$$\min_{\{W_i\}_{i=1}^m} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathcal{D}_j|} L_s(\mathcal{M}_i, \mathcal{D}_i, W_i), \tag{2}$$

where m denotes the number of devices.

On the server side, the vanilla FL aggregates all local models into a global one $\mathcal{M}_{global}$ which is parameterized by $\tilde{W}^g$.

In particular, it adopts a weighted average of the local model parameters $[W_i]_{i=1}^m$, i.e.,

$$\tilde{W}^g = \sum_{i=1}^{m} \frac{|\mathcal{D}_i|}{\sum_j |\mathcal{D}_j|} W_i, \tag{3}$$

which is the nearest center for all $\{W_i\}_{i=1}^m$ in terms of a weighted L2 distance:

$$\tilde{W}^g \in \arg\min_{\tilde{W}} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathcal{D}_j|} \|\tilde{W} - W_i\|_2^2. \tag{4}$$

More generally, we can replace the L2 distance in Eq. (4) by other distance metric $\mathrm{Dist}(\cdot,\cdot)$ and minimize the difference between the global model and all the local models, i.e.,

$$\min_{\tilde{W}} \frac{1}{m} \sum_{i=1}^{m} \mathrm{Dist}(W_i, \tilde{W}). \tag{5}$$

The above aims to find a consistent solution across global model and local models. Note that a direct macro average is used here regardless of the weight of each device, which treats every device equally. The weights used in Eq. (2) can easily be incorporated for a micro average.

The divergence $\mathrm{Dist}(\cdot,\cdot)$ between the global model and local models plays an essential role in the FL objective. The simple L2 distance for $\mathrm{Dist}(\cdot,\cdot)$ does not take into account the fact that two models can be identical under the arbitrary permutation of neurons in each layer. Hence, the lack of neuron matching may cause misalignment in that two neurons with similar functions and different indexes cannot be aligned across models [Yurochkin *et al.*, 2019]. However, the index-based neuron matching in FL [Shamir *et al.*, 2014] is the most widely used method and works well in various real applications. One potential reason for this is that the index-based neuron matching can also slowly align the function of neurons by repeatedly initializing all local models with the same global model. To simplify the description, we will discuss our method for index-based neuron matching, and then discuss a possible extension by adding function-based neuron matching [Wang *et al.*, 2020] (Section5.1).

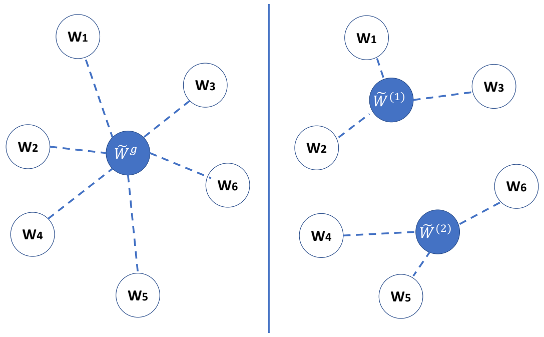

Figure 1: Comparison between single-center aggregation in vanilla FL (left) and multi-center aggregation in the proposed one (right). Each W_i represents the local model's parameters collected from the i-th device, which is denoted as a node in the space. $\tilde{W}$ represents the aggregation result of multiple local models.

3.2 Motivation

Federated learning (FL) usually aggregates all local models to a single global model. However, this single-center aggregation is fragile under heterogeneity. In contrast, we consider FL with multiple centers to better capture the heterogeneity by assigning nodes to different centers so only similar local models are aggregated. Consider two extreme cases for the number of centers, K: (1) when $K = 1$, it reduces to the FedAvg with a single global model, which cannot capture the heterogeneity and the global model might perform poorly on specific nodes; (2) When $K = m$, the heterogeneity problem can be avoided by assigning each node to one global model. But the data on each device used to update each global model can be insufficient and thus we lose the main advantage of FL. Our goal is to find a sweet point between these two cases to balance the advantages of federated averaging and the degradation caused by underlying heterogeneity.

Learning one unique model for each node has been discussed in some recent FL studies for better personalized models. They focus on making a trade-off between shared knowledge and professionalisation. The personalising strategy either applies fine-tuning of the global model [Zhao *et al.*, 2018] for each node, or only updates a subset of personalised layers for each node [Arivazhagan *et al.*, 2019; Liang *et al.*, 2020], or deploys a regularisation term in the objective [Deng *et al.*, 2020; Dinh *et al.*, 2020; Hanzely and Richtárik, 2020]. In contrast, Multi-center FL in this paper mainly focuses to address the heterogeneity challenge by assigning nodes to different global models during aggregation. But it can be easily incorporated in these peronalization strategies. In the following, we will start from the problem setting for the the vanilla FL, and then elucidate our motivation of improving FL's tolerance to heterogeneity by multi-center design.

4 Methodology

4.1 Multi-Center Model Aggregation

To overcome the challenges arising from the heterogeneity in FL, we propose a novel model aggregation method with multiple centers, each associating with a global model $\tilde{W}^{(k)}$ updated by aggregating a cluster of user's models with nearly IID data. In particular, all the local models will be grouped to K clusters, denoted as $C_1, \cdots, C_K$, each covering a subset of local models with parameters $\{W_j\}_{j=1}^{m_k}$.

An intuitive comparison between the vanilla FL and our multi-center FL is illustrated in Fig. 1. As shown in the left figure, there is only one center model in vanilla FL. In contrast, the multi-center FL shown in the right has two centers, $W^{(1)}$ and $W^{(2)}$, and each center represents a cluster of devices with similar data distributions and models. Obviously, the right one has a smaller intra-cluster distance than the left one. As discussed in the following Section 4, intra-cluster distance directly reflects the possible loss of the FL. Hence, a much smaller intra-cluster distance indicates our proposed approach potentially reduces the loss of FL.

4.2 Problem Formulation

Solving a joint optimization on a distributed network. The multi-center FL problem can be formulated as

$$\min_{\{W_i\},\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathcal{D}_j|} L_s(\mathcal{M}_i, \mathcal{D}_i, W_i) + \frac{\lambda}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \operatorname{Dist}(W_i, \tilde{W}^{(k)}), \tag{6}$$

where λ controls the trade-off between supervised loss and distance. We solve it by applying an alternative optimization between server and user: (1) on each node-i, we optimize the above objective w.r.t. W_i while fixing all the other variables; and (2) on the server, we optimize $\{r_i^{(k)}\}, \{\tilde{W}^{(k)}\}$ for $i \in [m]$ and $k \in [K]$ while fixing all local models $\{W_i\}$.

Multi-center assignment at the server end. The second term in Eq. (6) aims to minimize the distance between each local model and its nearest global model. Under the non-IID assumption, the data located at different devices can be grouped into multiple clusters where the on-device data in the same cluster are likely to be generated from one distribution. As illustrated on the right of Fig. 1, we optimizes the assignments and global models by minimizing the intra-cluster distance, i.e,

$$\min_{\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \operatorname{Dist}(W_i, \tilde{W}^{(k)}), \tag{7}$$

where cluster assignment $r_i^{(k)}$, as defined in Eq. (9), indicates whether device-i belongs to cluster-k, and $\tilde{W}^{(k)}$ is the parameters of the aggregated model for cluster-k.

Distance-constrained loss for local model optimization. Because the distance between the local model and the global model are essential to our new loss, we don't expect the local model will be changed too much during the local updating stage. The new loss consists of a supervised learning loss and a regularization term to constrain the local model to ensure it is not too far from the global model. This kind of regularization term is also known as the proximal term in [Li *et al.*, 2018] that can effectively limit the impact of the variable local updates in FL. We minimize the loss below for each local model W_i as follows:

$$\min_{W_i} \frac{|D_i|}{\sum_j |\mathcal{D}_j|} \cdot L_s(\mathcal{M}_i, \mathcal{D}_i, W_i) + \frac{\lambda}{m} \sum_{k=1}^{K} r_i^{(k)} \operatorname{Dist}(W_i, \tilde{W}^{(k)}) \tag{8}$$

4.3 Optimization Algorithm

In general, Expectation-Maximization (EM) [Bishop, 2006] can be used to solve the distance-based objective function of clustering, e.g., K-Means. However, in contrast to the general objective of clustering, our proposed objective, as described in Eq. 7, has a dynamically changing W_i during optimization. Therefore, we adapt the Stochastic Expectation Maximization (SEM) [Cappé and Moulines, 2009] optimization framework by adding one step, i.e., updating W_i. In the modified SEM optimization framework, named federated SEM (FeSEM), we sequentially conduct: 1) E-step – updating cluster assignment $r_i^{(k)}$ with fixed W_i, 2) M-step – updating cluster centers $\tilde{W}^{(k)}$, and 3) updating local models by providing new initialization $\tilde{W}^{(k)}$.

Algorithm 1: FeSEM – Federated Stochastic EM

```
Initialize $K$, $\{W_i\}_{i=1}^m$, $\{\tilde{W}^{(k)}\}_{k=1}^K$
while stop condition is not satisfied do
    E-Step:
    Calculate distance $d_{ik} \leftarrow \operatorname{Dist}(W_i, \tilde{W}^{(k)})\ \forall i, k$
    Update $r_i^{(k)}$ using $d_{ik}$ (Eq. 9)
    M-Step:
    Group devices into $C_k$ using $r_k^{(k)}$
    Update $\tilde{W}^{(k)}$ using $r_i^{(k)}$ and $W_i$ (Eq. 10)
    for each cluster $k = 1, \dots K$ do
        for $i \in C_k$ do
            Send $\tilde{W}^{(k)}$ to device $i$
            $W_i \leftarrow$ Local_update$(i, \tilde{W}^{(k)})$
        end
    end
end
```

Firstly, for the **E-Step**, we calculate the distance between the cluster center and nodes – each node is the model's parameters W_i, then update the cluster assignment $r_i^{(k)}$ by

$$r_i^{(k)} = \begin{cases} 1, & if\ k = \arg\min_j \operatorname{Dist}(W_i, \tilde{W}^{(j)}) \\ 0, & otherwise. \end{cases} \tag{9}$$

Secondly, for the **M-Step**, we update the cluster center $\tilde{W}^{(k)}$ according to the W_i and $r_i^{(k)}$, i.e.,

$$\tilde{W}^{(k)} = \frac{1}{\sum_{i=1}^{m} r_i^{(k)}} \sum_{i=1}^{m} r_i^{(k)} W_i. \tag{10}$$

Thirdly, to **update the local models**, the global model's parameters $\tilde{W}^{(k)}$ are sent to each device in cluster k to update its local model, and then we can fine-tune the local model's parameters W_i using a supervised learning algorithm on its own private training data while considering the new loss as described in Eq. 8.

The local training procedure is a supervised learning task by adding a distance-based regularization term. The local model is initialized by the global model $\tilde{W}^{(k)}$ which belong to the cluster associated with the node.

Lastly, we repeat the three stochastic updating steps above until convergence. The sequential executions of the three updates comprise the iterations in FeSEM's optimization procedure. In particular, we sequentially update three variables $r_i^{(k)}$, $\tilde{W}^{(k)}$, and W_i while fixing the other factors. These three variables are jointly used to calculate the objective of our proposed multi-center FL in Eq. 7.

We implement FeSEM in Algorithm 1 which is an iterative procedure. As elaborated in Section 4.2, each iteration

```
Algorithm 2: Local_update
  Input: $i$ – device index
    $\tilde{W}^{(k)}$ – the model parameters from server
  Output: $W_i$ – updated local model
  Initialization: $W_i \leftarrow \tilde{W}^{(k)}$
  for $N$ local training steps do
    Update $W_i$ with training data $\mathcal{D}_i$ (Eq. 8)
  end
  Return $W_i$ to server
```

comprises of three steps to update the cluster assignment, the cluster center, and the local models, respectively. In the third step to update the local model, we need to fine-tune the local model by implementing Algorithm 2.

5 Some Possible Extensions

To further handle heterogeneous data in FL scenario, our multi-center FL approach can be easily extended with other packages. We discuss two beneficial techniques here.

5.1 Model Aggregation with Neuron Matching

The vanilla FL algorithm, FedAvg [McMahan *et al.*, 2017], uses model aggregation with index-based neuron matching which may cause the incorrect alignmentment. Neurons with similar functions are usually take different indexes in two models. Recently, a function-based neuron matching [Wang *et al.*, 2020] in FL is proposed to align two models by matching the neurons with similar functions. In general, the index-based neuron matching can gradually align the neuron's functionality across nodes by repeatedly forcing each local model to be initialized using the same global model. However, the function-based neuron matching can speed up the convergence of neuron matching and preserve the unique functional neuron of the minority groups.

In this work, we integrate layer-wise matching and then averaging(MA) [Wang *et al.*, 2020] into ours to increase the capacity to handle heterogeneous challenges. The distance between the local model and the global model is the neuron matching score that is calculated by estimating the maximal posterior probability of the j-th client neuron l generated from a Gaussian with mean W_i, and ϵ and $f(\cdot)$ are guided by the Indian Buffet Process prior [Yurochkin *et al.*, 2019].

5.2 Selection of K

The selection of K, the number of centers, is essential for a multi-center FL. In general, the K is defined based on the prior experience or knowledge of data. If there is no prior knowledge, the most straightforward solution is to run the algorithm using different K and then select the K with the best performance in terms of accuracy or intra-cluster distance. Selecting the best K in a large-scale FL system is time consuming, hence we simplify the process by running the algorithm on a small number of sampled nodes with several communication rounds. For example, we can randomly select 100 nodes and test K in FL with three communication rounds only, and then apply the K to the large-scale FL.

6 Experiments

As a proof-of-concept scenario to demonstrate the effectiveness of the proposed method, we experimentally evaluate and analyze FeSEM on two datasets.

6.1 Training Setups

Datasets. We employed two publicly-available federated benchmarks datasets introduced in LEAF [Caldas *et al.*, 2018]. LEAF is a benchmarking framework for learning in federated settings. The datasets used are Federeated Extended MNIST (FEMNIST)[1] [Cohen *et al.*, 2017] and Federated CelebA (FedCelebA)[2] [Liu *et al.*, 2015]. We follow the setting of the benchmark data in LEAF. In FEMNIST, images is split according to the writers. For FedCelebA, images are extracted for each person and developed an on-device classifier to recognize whether the person smiles or not. A statistical description of the datasets is described in Table 2.

Local model. We use a CNN with the same architecture from [Liu *et al.*, 2015]. Two data partition strategies are used: (a) an ideal IID data distribution using randomly shuffled data, (b) a non-IID partition by use a $\mathbf{p}_k \sim Dir_J(0.5)$. Part of the code is adopted from [Wang *et al.*, 2020]. For FEMINST data, the local learning rate is 0.003 and epoch is 5. and for FedCelebA, 0.03 and 10 respectively.

Baselines. In the scenario of solving statistical heterogeneity, we choose FL methods as follows:

1. **NonFed**: We will conduct the supervised learning task at each device without the FL framework.
2. **FedSGD**: uses SGD to optimise the global model.
3. **FedAvg**: is an SGD-based FL with weighted averaging. [McMahan *et al.*, 2017] .
4. **FedCluster**: is to enclose FedAvg into a hierarchical clustering framework [Sattler *et al.*, 2019].
5. **HypoCluster(K)**: is a hypothesis-based clustered-FL algorithm with different K [Mansour *et al.*, 2020].
6. **Robust** our implementations based on the proposed method in [Ghosh *et al.*, 2019], see this baseline settings in Appendix.
7. **FedDANE**: this is an FL framework with a Newton-type optimization method. [Li *et al.*, 2019c].
8. **FedProx**: this is our our own implementations following [Li *et al.*, 2018]. We set scaler of proximal term to 0.1.
9. **FedDist**: we adapt a distance based-objective function in Reptile meta-learning [Nichol and Schulman, 2018] to a federated setting.
10. **FedDWS**: a variation of FedDist by changing the aggregation to weighted averaging where the weight depends on the data size of each device.
11. **FeSEM**(K): our multi-center FL implemented on federated SEM with K clusters.
12. **FeSEM-MA**(K): FeSEM integrates the matched averaging [Wang *et al.*, 2020].

[1] http://www.nist.gov/itl/products-and-services/emnist-dataset
[2] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Datasets	FEMNIST				FedCelebA			
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	79.0±2.0	67.6±0.6	81.3±1.9	51.0±1.2	83.8±1.4	66.0±0.4	83.9±1.6	67.2±0.6
FedSGD	70.1±2.2	61.2±3.4	71.5±1.8	46.7±1.2	75.7±2.3	60.7±2.4	75.6±2.0	55.6±2.6
FedAvg	84.9±2.0	67.9±0.4	84.9±1.6	45.4±1.9	86.9±0.5	**78.0**±1.0	86.1±0.4	54.2±0.6
FedDist	79.3±0.8	67.5±0.5	79.8±1.1	50.5±0.5	71.8±0.9	61.0±0.8	71.6±1.0	61.1±0.7
FedDWS	80.4±0.8	67.2±1.6	80.6±1.2	51.7±1.1	73.4±1.7	59.3±0.9	73.4±1.9	50.3±0.5
Robust(TKM)	78.4±1.0	53.1±0.5	77.6±0.7	53.6±0.7	90.1±1.3	68.0±0.7	90.1±1.3	68.3±1.1
FedCluster	84.1±1.1	64.3±1.3	84.2±1.0	**64.4**±1.6	86.7±0.7	67.8±0.9	87.0±0.9	67.8±1.3
HypoCluster(3)	82.5±1.7	61.3±0.6	82.2±1.3	61.6±0.9	76.1±1.5	53.5±1.0	72.7±1.8	53.8±1.9
FedDane	40.0±2.9	31.8±3.1	41.7±2.4	31.7±1.6	76.6±1.1	61.8±2.0	75.9±1.0	62.1±2.2
FedProx	72.6±1.8	62.8±1.6	74.3±2.1	50.6±1.2	83.8±2.0	60.9±1.2	84.9±1.8	65.7±1.2
FeSEM(2)	84.8±1.1	65.5±0.4	84.8±1.6	52.0±0.5	89.1±1.3	64.6±1.0	89.0 ±1.3	56.0±1.3
FeSEM(3)	87.0±1.2	68.5±2.0	86.9±1.2	41.7±1.5	88.1±1.9	64.3±0.8	87.5±2.0	55.9±0.8
FeSEM(4)	**90.3**±1.5	70.6±0.9	**91.0**±1.8	53.4±0.6	**93.6**±2.7	**74.8**±1.5	**94.1**±2.2	**69.5**±1.1
FeSEM-MA(3)	**90.4**±1.5	**71.4**±0.5	87.0±2.0	64.3±0.5	84.5±0.8	64.1±0.7	85.1±1.0	63.0±1.3

Table 1: Comparison of our proposed FeSEM(K) algorithm with the baselines on FEMNIST and FedCelebA datasets. Note the number in parenthesis following "FeSEM" denotes the number of clusters, K.

DATASET	FEMNIST	FedCelebA
# of data points	805,263	200,288
# of device	3,550	9,343
# of Classes	62	2
Model architecture	CNN	CNN

Table 2: Statistics of datasets.

Training settings. We used 80% of each device's data for training and 20% for testing. For the initialization of the cluster centers in FeSEM, we conducted pure clustering 20 times with randomized initialization, and then the "best" initialization, which has the minimal intra-cluster distance, was selected as the initial centers for FeSEM. For the local update procedure of FeSEM, we set N to 1, meaning we only updated W_i once in each local update.

Evaluation metrics. Given numerous devices, we evaluated the overall performance of the FL methods. We used classification accuracy and F1 score as the metrics for the two benchmarks. In addition, due to the multiple devices involved, we explored two ways to calculate the metrics, i.e., micro and macro. The only difference is that when computing an overall metric, "micro" calculates a weighted average of the metrics from devices where the weight is proportional to the data amount, while "macro" directly calculates an average over the metrics from devices.

6.2 Experimental Study

Comparison study. As shown in Table 1, we compared our proposed FeSEM with the baselines and found that FeSEM achieves the best performance in most cases. But, it is observed that the proposed model achieves an inferior performance for Micro F1 score on the FedCelebA dataset. A possible reason for this is that our objective function defined in Eq. 7 does not take into account the device weights. Hence, our model is able to deliver a significant improvement in terms of "macro" metrics. Furthermore, as show in the last three columns in Table 1, we found that FeSEM with a larger number of clusters empirically achieves a better performance, which verifies the correctness of the non-IID assumption of the data distribution.

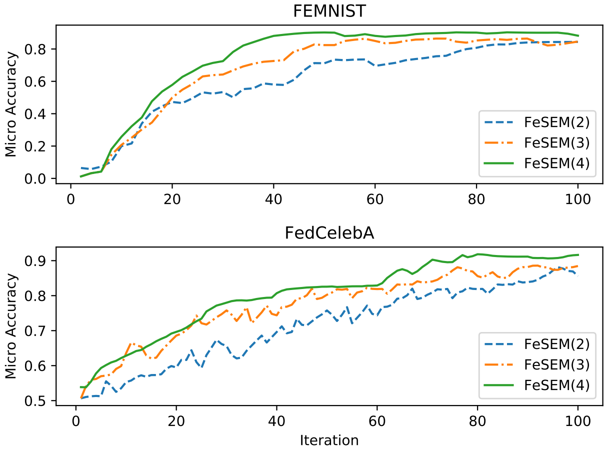

Figure 2: Convergence analysis for the proposed FeSEM with different cluster number (in parenthesis) in terms of micro-accuracy.

Convergence analysis. To verify the convergence of the proposed approach, we conducted a convergence analysis by running FeSEM with different cluster numbers K (from 2 to 4) in 100 iterations. As shown in Fig. 2, FeSEM can efficiently converge on both datasets and it can achieve the best performance with the cluster number $K = 4$.

Clustering analysis. To check the effectiveness of our proposed optimization method and whether the devices grouped into one cluster have similar model, we conducted a clustering analysis via an illustration. We used two-dimensional figures to display the clustering results of the local models derived from FeSEM(4) on the FEMNIST dataset. In particular, we randomly chose 400 devices from the dataset and plotted each device's local model as one point in the 2D space after PCA dimension reduction. As shown in Fig. 3, the dataset

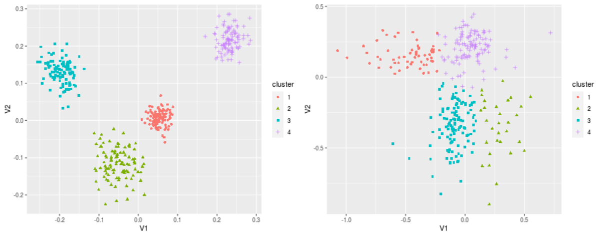

Figure 3: Clustering analysis for different local models (using PCA) derived from FeSEM(4) using FEMNIST and Celeba data.

suitable for four clusters that are distinguishable to each other.

Case study on clustering. To intuitively judge whether nodes grouped into the same cluster have a similar data distribution, we conducted case studies on a case of two clusters that are extracted from a trained FeSEM(2) model. For FMNIST, as shown on the top of Fig. 4, cluster on the right consists writers who are likely to recognize hand-writings with a smaller font, and on the left consists writers who are likely to recognize hand-writing with a bolder and darker font. For FedCelebA, see full face images in Appendix section 2, the face recognition task in cluster1 is likely to handle the smiling faces with a relatively simple background, also exhibits to be young people. While cluster on the right is likely to handle the faces with more diverse background and also seems to be more older people.

7 Case study

As shown in Figure 4, there are two clusters for FEMNIST and CIFAR-10 datasets respectively. The upper part shows the clustering effect of FeSEM on dataset MINIST by writers, on the left are three writers handwritten digits which are smaller and lighter those the right. The botttom part shows the clustering of CIFAR-10 data with 10 classes in which one class is about people. In the class of people, our algorithm finds one cluster to represent young beauties and another clustter to represent the aging people.

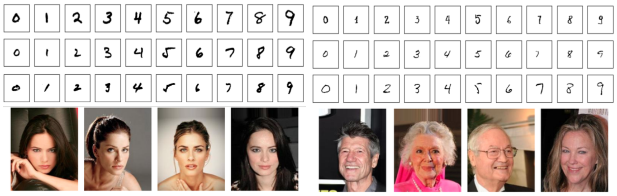

Figure 4: Cluster demonstrations on FEMINIST and CIFAR-10 (people)

8 Multi-Center FL with Neuron Matching

Applying neuron matching on FL could enhance its performance by providing a fine-grained function-based alignment of neurons across two models. A neuron matching-enhanced multi-center FL, namely FeSEM-MA, are implemented and compared with FeSEM. Because the benchmark of neuron matching [Wang *et al.*, 2020] uses MNIST and CIFAR-10 with specific pre-processing. As the selected datasets and pre-processing are different to other baseline algorithms, we only compare the **FeSEM** with the neuron matched averaging(MA) enhanced version **FeSEM-MA** in terms of $K = 2$. As demonstrated in the comparison result (Table 3), the FeSEM-MA can improve the performance on MNIST data.

Datasets	MNIST		CIFAR-10	
Metrics(%)	Micro-Acc	Micro-F1	Micro-Acc	Micro-F1
FeSEM(3)	85.7	87.3	59.3	52.8
FeSEM-MA(3)	95.6	94.2	61.0	60.0

Table 3: Comparison of FeSEM with or without neuron matching

9 Conclusion

In this work, we proposed a novel FL algorithm to tackle the non-IID challenge of FL. This proposed method can efficiently capture the multiple hidden distributions of numerous devices or users. An optimization approach, federated SEM, is also proposed to solve the multi-center FL problem effectively. The experimental results show the effectiveness of our algorithm, and several analyses are further provided for an deeper insight into the proposed approach.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv:1912.00818*, 2019.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[Bonawitz *et al.*, 2019] Keith Bonawitz, Hubert Eichner, and et.al. Towards federated learning at scale: System design. *ArXiv*, abs/1902.01046, 2019.

[Caldas *et al.*, 2018] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv:1812.01097*, 2018.

[Cao *et al.*, 2020] Tien-Dung Cao, Tram Truong-Huu, Hien Tran, and Khanh Tran. A federated learning framework for privacy-preserving and parallel training. *arXiv:2001.09782*, 2020.

[Cappé and Moulines, 2009] Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society*, 71(3):593–613, 2009.

[Cohen *et al.*, 2017] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[Deng *et al.*, 2020] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv:2003.13461*, 2020.

[Dinh *et al.*, 2020] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv:2006.08848*, 2020.

[Geyer *et al.*, 2017] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv:1712.07557*, 2017.

[Ghosh *et al.*, 2019] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv:1906.06629*, 2019.

[Ghosh *et al.*, 2020] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *arXiv:2006.04088*, 2020.

[Haddadpour and Mahdavi, 2019] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv:1910.14425*, 2019.

[Hanzely and Richtárik, 2020] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv:2002.05516*, 2020.

[Hsu *et al.*, 2019] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv:1909.06335*, 2019.

[Jeong *et al.*, 2018] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv:1811.11479*, 2018.

[Kairouz *et al.*, 2019] Peter Kairouz, H Brendan McMahan, and et.al. Advances and open problems in federated learning. *arXiv:1912.04977*, 2019.

[Konecný *et al.*, 2018] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv:1610.05492*, 2018.

[Li and Wang, 2019] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv:1910.03581*, 2019.

[Li *et al.*, 2018] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv:1812.06127*, 2018.

[Li *et al.*, 2019a] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI*, 2019.

[Li *et al.*, 2019b] Qinbin Li, Zeyi Wen, and Bingsheng He. Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv:1907.09693*, 2019.

[Li *et al.*, 2019c] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *ACSSC*, pages 1227–1231. IEEE, 2019.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv:2001.01523*, 2020.

[Lim *et al.*, 2020] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor*, 2020.

[Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *arXiv:2006.07242*, 2020.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[Liu *et al.*, 2019] Yang Liu, Zhuo Ma, Ximeng Liu, Zhuzhu Wang, Siqi Ma, and Ken Ren. Revocable federated learning: A benchmark of federated forest. *arXiv:1911.03242*, 2019.

[Long *et al.*, 2020] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pages 240–254. Springer, 2020.

[Long *et al.*, 2021] Guodong Long, Tao Shen, Yue Tan, Leah Gerrard, Clarke Allison, and Jing Jiang. Federated learning for privacy-preserving open innovation future on digital health. In *Humanity Driven AI*. Springer, 2021.

[Luo *et al.*, 2019] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. Real-world image datasets for federated learning. *arXiv:1910.11089*, 2019.

[Lyu *et al.*, 2020] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv:2003.02133*, 2020.

[Mansour *et al.*, 2020] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv:2002.10619*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Nichol and Schulman, 2018] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv:1803.02999*, 2, 2018.

[Rieke *et al.*, 2020] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[Rouhani *et al.*, 2018] Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. Deepsecure: Scalable provably-secure deep learning. In *The 55th Annual Design Automation Conference*, page 2. ACM, 2018.

[Sattler *et al.*, 2019] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. *arXiv:1910.01991*, 2019.

[Shamir *et al.*, 2014] Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *ICML*, pages 1000–1008, 2014.

[Tan *et al.*, 2021] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning over heterogeneous devices. *arXiv preprint arXiv:2105.00243*, 2021.

[Wang *et al.*, 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020.

[Xu *et al.*, 2021] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *TIST*, 2019.

[Yang *et al.*, 2020] Chengxu Yang, QiPeng Wang, Mengwei Xu, Shangguang Wang, Kaigui Bian, and Xuanzhe Liu. Heterogeneity-aware federated learning. *arXiv:2006.06983*, 2020.

[Yu *et al.*, 2020] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Heterogeneous federated learning. *arXiv:2008.06767*, 2020.

[Yurochkin *et al.*, 2019] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. *arXiv:1905.12022*, 2019.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv:1806.00582*, 2018.