Machine Learning
https://doi.org/10.1007/s10994-023-06443-5

Communication-efficient clustered federated learning via model distance

Mao Zhang[1,3] · Tie Zhang[1,3] · Yifei Cheng[2,3] · Changcun Bao[4] · Haoyu Cao[4] · Deqiang Jiang[4] · Linli Xu[1,3]

Received: 6 June 2023 / Revised: 28 August 2023 / Accepted: 7 October 2023

Abstract
Clustered Federated Learning (CFL) leverages the differences among data distributions on clients to partition all clients into several clusters for personalized federated training. Compared with the conventional federated algorithms such as FedAvg, existing methods for CFL require either more communication costs or multi-stage computation overheads. In this paper, we propose an iterative CFL framework with almost the same communication cost as FedAvg in each round based on a novel model distance. Specifically, the model distance measures the discrepancy between the client model and the cluster model so that we can estimate the cluster identities for clients on the server side. The proposed model distance considers class-wise model dissimilarity, which enables us to apply it to multi-class classification even when the labels are non-iid across clients. To calculate the proposed model distance, we introduce two sampling methods which generate samples from feature distributions approximately without accessing the raw dataset. Experimental results show that our method can achieve superior and comparable performance on non-iid and iid data respectively with less communication cost compared with the baselines.

Keywords Federated learning · Client clustering · Data heterogeneity · Model distance

1 Introduction

Federated Learning (FL) is a novel distributed machine learning paradigm where multiple users or mobile devices collaboratively learn a model under privacy constraints (McMahan et al., 2017; Li et al., 2019; Li and Sahu, 2020; Kairouz et al., 2019; Karimireddy et al., 2020). In this scenario, the training data is distributed on a large number of clients, and there is a central parameter server which is responsible for coordinating the training process of the entire system. Clients are not allowed to send their own private raw data and the relevant statistical information to the parameter server or other clients due to the risk of privacy leaking. In general, FL assumes that we can train a single global model that is able to fit data distributions for all clients. However, this assumption is hard to guarantee

Editor: Vu Nguyen, Dani Yogatama.

Extended author information available on the last page of the article

in practical distributed training tasks as the variations caused by different regions, ages and genders often lead to non-iid (independent and identically distributed) data distributions across clients (Sattler and Müller, 2020). Moreover, the limited computation and storage capacity of current mobile devices make it infeasible to deploy a sufficiently large model to obtain desirable performance. Hence, various personalized federated learning algorithms have recently been proposed in order to train a personalized model for each client while communicating with the server or other clients.

As one class of personalized federated learning methods, Clustered Federated Learning (CFL) (Sattler and Müller, 2020; Ghosh et al., 2019, 2020) explicitly considers the underlying cluster structure of client populations, which commonly assumes that there are K different global data (feature) distributions and the local dataset on each client is generated from one of them. The underlying distribution discrepancies could be leveraged to partition the clients to K clusters, where the clients within each cluster have the same or similar feature distributions referring to the personal interests and preferences. As a consequence, each cluster can be viewed as an independent machine learning task and trained with conventional single-model federated leaning algorithms. In general, CFL scenarios can be divided into two categories. (1) Multi-source data: the local datasets on clients are drawn from different devices or regional styles. (2) Incongruent data: clients have different opinions (labels) regarding the same data (Sattler and Müller, 2020).

Recently, some attempts have been made to achieve model clustering for clients in the federated learning setting (Sattler and Müller, 2020; Ghosh et al., 2020; Fu et al., 2021). The first challenge arising here is that the only information accessible to the server during each communication round is the parameters of local models and cluster models, which may incur high communication and related computation costs for clients. Secondly, while in some works (Ghosh et al., 2019) the Euclidean distance between model parameters has been used to cluster the clients via a multi-stage training method, a shorter Euclidean distance does not always mean a pair of more similar function mappings due to overparameterization and permutation invariance of modern neural networks (Wang and Yurochkin, 2020). The issue becomes more serious when the label distributions of local datasets are different across clients (i.e., **label non-iid**) even though the sources of feature distributions are same. Therefore, the measure of model dissimilarity is still an essential problem to be further studied. To tackle the challenges described above, in this paper we consider measuring the dissimilarity between cluster models and client models on the server side from the perspective of model distance for multi-class classification problems.

Model distance is originally proposed for the fault diagnosis problem and defined in the integral form. The model distance is supposed to be shorter if the outputs of two models are similar for a given input x. In this paper, we propose a novel model distance for cluster-client model pairs in CFL which is defined by the integral over the corresponding cross distributions, which are constructed based on the feature distributions of clusters and the label distributions of clients. The new model distance can be estimated on the server side without extra communication, thus leading to $1/K$ communication cost for downloading models compared to the existing iterative method IFCA (Ghosh et al., 2020). Moreover, the structure of the new model distance captures the class-wise dissimilarities, making it effective even if the local dataset on a client contains only partial classes. To compute the new model distance, two sampling methods depending only on model parameters are proposed to generate samples from the corresponding cross distributions approximately, which will be used to estimate the model distance in the form of finite sum. Finally, we propose a novel communication-efficient framework for CFL named MD-ICFL (Model Distance based Iterative Clustered Federated Learning). To validate the effectiveness of the

proposed method, MD-ICFL is tested and compared with baselines in multiple different CFL scenarios. The experimental results show that the proposed framework can achieve better clustering performance with less communication cost, demonstrating the superiority of the new model distance and sampling methods.

The contributions of our work are summarized as follows:

- We design a novel and effective model distance which is robust to label non-iid scenarios to measure the dissimilarity of cluster-client model pairs in CFL.
- We introduce two sampling methods to generate samples from the corresponding cross distributions to estimate the performance of the proposed model distance.
- We propose a novel communication-efficient framework MD-ICFL for CFL, which performs client clustering on the server side based only on model parameters.

2 Related works

2.1 Personalized federated learning

Due to data heterogeneity in realistic scenarios, various personalized federated learning methods have been proposed to achieve personalization for clients instead of training a single global model. Google first introduces a framework (Wang et al., 2019) that fine-tunes the global model locally after receiving it from the server to achieve personalized language models in smart phones. Other lines of research improve model aggregation by designing varying weights for different clients according to the similarity between clients. For example, FedAMP (Huang et al., 2021) leverages an attention-inducing function to measure the difference between the data distributions. FedFoMo (Zhang et al., 2020) considers the first-order optimal aggregation weights based on the personalized loss functions on clients and achieves the SOTA performance for most of the scenarios with data heterogeneity in FL. In addition, MOCHA (Smith et al., 2017) formulates federated learning as a multi-task learning setting where clients can naturally be viewed as multiple tasks which have different objectives while collaborating with each other.

2.2 Clustered federated learning

Many works have been proposed to explore client clustering in the context of CFL. Among them, Ghosh et al. (2019) performs a multi-phase training procedure where all clients firstly train their models locally until convergence, which are then clustered based on the Euclidean distance between model parameters, after that, the cluster partition will be used to employ conventional FL in each cluster. Similarly, PFA (Liu et al., 2021) also proposes a multi-phase method for client clustering based on the sparsity of the outputs of Relu activations in neural networks. Considering the incongruent data issue, Sattler and Müller (2020) introduces a hierarchical clustering method based on the cosine similarity between gradients. All the above methods of model clustering need to pre-train the global or local models in advance until loss functions converge approximately, which is therefore computationally expensive and not suitable for real-time, large-scale FL training scenarios. Another line of research estimates the cluster identities of clients during collaborative training in an iterative way. For instance, IFCA (Ghosh et al., 2020) measures the dissimilarity between cluster models and client models by comparing the values of the loss function computed on the client side after the clients receive the cluster models. As a consequence, while being

effective and theoretically guaranteed, IFCA requires K times communication cost of the FedAvg framework in each round. In contrast, our proposed method is able to avoid this by using a new model distance which can be estimated on the server side. Recently, CIC-FL (Fu et al., 2021), the most relevant work to ours, focuses on the label non-iid setting and constructs features that are sensitive to concept shift but robust to class imbalance for each client, based on which the clients are then bipartitioned recursively. Nevertheless, issues may arise when a client lacks the data of some classes, as it is unable to obtain the label-wise gradients corresponding to those unseen classes, which are required when constructing the client features.

3 Clustered federated learning with model distance

3.1 Problem formulation

We follow the standard CFL paradigm with one center *server* machine and N *client* machines. There are K different global conditional distributions, i.e., **global feature distributions** $\varphi_j(x|y), j \in [K]$, where x and y denote the feature representation and the corresponding label respectively. The i-th client, $i \in [N]$, trains its C-class classification model using the local private dataset $\mathcal{D}_i$ generated from the joint distribution $p_i(x, y)$, where the label distribution is $p_i(y)$ and the conditional distribution $p_i(x|y)$ is one of K global feature distributions correspondingly. In addition to the conditional distribution, the label distribution $p_i(y)$ could also be non-identical across all clients. Denoting the cluster identity of client i as $s_i \in [K]$, our goal is to group N clients into K disjoint clusters $S_1^*, ..., S_K^*$ based on $\{s_i\}$ correctly so that the clients within each cluster share the same conditional distribution and we can learn one personalized model for each cluster. In other words, we view every ground-truth cluster as an independent task that minimizes a global loss function $F_j(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_j^*}\left[f(\theta;x, y)\right]$ for all $j \in [K]$, where $f(\theta;x, y)$ is the loss function associated with the instance (x, y) and the model parameter θ. $\mathcal{D}_j^* := \varphi_j(x, y)$ is the global joint distribution of feature x and label y in the j-th cluster determined by the populations of clients in this cluster. We call the settings with non-identical conditional (feature) distributions and label distributions as **feature non-iid** and **label non-iid** respectively.

3.2 Federated model distance for client clustering

In clustered federated learning, the primary goal of client clustering is to group the clients based on the sources of their local datasets, i.e., the global feature distributions $\varphi_j(x|y), j \in [K]$. In order to identify which global feature distribution each client belongs to without being interfered by the unbalanced label distributions, for a given client i whose local joint distribution is $p(x, y) = p(x|y)p(y)$ (the subscript i is omitted for simplicity), we construct K cross distributions $Q_j := q_j(x, y) = \varphi_j(x|y)p(y), j \in [K]$ based on the label distribution $p(y)$ and the K global feature distributions. Obviously, the dissimilarity among these K cross distributions results only from different global feature distributions. Therefore, we can estimate the cluster identity of a client by comparing the distribution distance between the joint distribution $p(x, y)$ and $q_j(x, y)$ directly. Ideally, the distribution distance between $q_j(x, y)$ and $p(x, y)$ would be zero if the client belongs to cluster j according to the definition in Sect. 3.1.

Motivated by the above discussion, we first measure the distance between the local joint distribution $p(x, y)$ and K cross distributions $q_j(x, y)$ with the Total Variation (TV) distance, which is a metric commonly used to measure the dissimilarity between two distributions. The reason we choose the TV distance instead of other metrics like the KL convergence is that the TV distance is defined with the absolute value (L_1 norm when the sample space is discrete) and thus robust to incorrect probability estimation. Formally, the distance between $p(x, y)$ and $q_j(x, y)$ can be written as

$$\frac{1}{2}\int_{x,y} \mid p(x,y) - q_j(x,y) \mid \mathrm{d}x\, \mathrm{d}y. \tag{1}$$

If the computation of this distance could be performed on the server side, we can use an EM-type iterative process to achieve communication-efficient clustered federated learning where clients require a communication cost of $O(|\theta|)$ in each communication round as FedAvg. $|\theta|$ denotes the amount of client model parameters. In other words, clients can use only $1/K$ communication cost for downloading models compared to the iterative algorithm IFCA (Ghosh et al., 2020).

Nevertheless, the computation of the above distance is not tractable as the real distributions $p(x, y)$ and $q_j(x, y)$ are both unknown to us and we are not permitted to estimate them with the raw data on the clients due to privacy constraints either. To tackle this problem, we show that the TV distance of $p(x, y)$ and $q_j(x, y)$ can be bounded by the sum of TV distances of the posterior distributions and the prior distributions, based on which we derive our novel distance. Specifically, we have

$$\begin{aligned}
&\frac{1}{2}\int_{x,y} \mid p(x,y) - q_j(x,y) \mid \mathrm{d}x\mathrm{d}y \\
=\ &\frac{1}{2}\int_{x,y} \mid p(y|x)p(x) - q_j(y|x)q_j(x) \mid \mathrm{d}x\mathrm{d}y \\
\leq\ &\underbrace{\frac{1}{2}\mathop{\mathbb{E}}_{x\sim Q_j}\Big[\int_y \mid p(y|x) - q_j(y|x) \mid \mathrm{d}y\Big]}_{B_1} \\
&+ \underbrace{\frac{1}{2}\int_x \mid p(x) - q_j(x) \mid \mathrm{d}x}_{B_2}
\end{aligned} \tag{2}$$

where the first part in the upper bound B_1 is the expectation of the TV distance between the posterior distributions $p(y|x)$ and $q_j(y|x)$ for x over the cross distribution Q_j. The second term B_2 is the TV distance between two prior distributions. Both of them are upper bounded by 1 for any j. It is worth noting that B_2 would be 0 if the "incongruent data" issue occurs with uniform label distribution $p(y)$. The reason is that the difference between two global feature distributions comes from only the different opinions regarding the same data and thus the prior distribution $p(x)$ and $q_j(x)$ are the same. While in the "multi-source data" scenario, B_1 and B_2 in (2) have similar trends in value for every j. In both cases, we can use B_1 as a surrogate to compare the TV distances between $p(x, y)$ and different $q_j(x|y)$ approximately.

We also note that for a given x, the term $\int_y \mid p(y|x) - q_j(y|x) \mid \mathrm{d}y$ in B_1 is the TV distance between two posterior distributions which can be viewed as the probability output

of a discriminative model, i.e., the softmax vector. This indicates the TV distance between the posterior distributions in B_1 can naturally be estimated by the distance between the outputs of two models. Therefore, we transform the measurement of distribution distances between $p(x, y)$ and $q_j(x, y)$ to that of the TV distance between the posterior distributions. Formally, we define a novel metric based on the conventional model distance (Chen et al., 2013) for CFL.

Definition 1 Given a client model θ_i which could be trained on the label non-iid dataset $\mathcal{D}_i$ and a cluster model θ_j^* corresponding to a global feature distribution $\varphi_j(x|y)$, the federated model distance d_{ij} between θ_j^* and θ_i is defined as

$$d_{ij} = \int_x \|h(\theta_i;x) - h(\theta_j^*;x)\|_1 \mathrm{d}\mu(x) \tag{3}$$

$\mu(x)$ is the probability density function over the cross distribution Q_{ij}, where the label distribution is that of the dataset $\mathcal{D}_i$ on client i notated by $p_i(y)$ and the feature distribution is the global feature distribution $\varphi_j(x|y)$. $h(\theta, x)$ represents the probability (softmax) output of the model θ on the input x and $\|\cdot\|_1$ is the L_1 norm.

To be more specific, we regard the federated model distance d_{ij} as an estimation of the B_1 term in the inequality (2). Due to the structure of the cross distribution Q_{ij}, we can rewrite d_{ij} as the sum of C terms of different classes with the expectation form

$$\begin{aligned} d_{ij} &= \sum_{k=1}^{C} p_i(y=k) \cdot \mathop{\mathbb{E}}_{x\sim\varphi_j(x|y=k)} \left[\|h(\theta_i;\mathbf{x}) - h(\theta_j^*;\mathbf{x})\|_1\right] \\ &= \sum_{k=1}^{C} p_i(y=k) \cdot d_{ij}^k. \end{aligned} \tag{4}$$

We define the above expectation term d_{ij}^k as the **class-wise model distance** representing the dissimilarity between the cluster-client model pair (j, i) in the k-th class. Since d_{ij}^k is defined in the form of expectation, we estimate them by the finite sum of the TV distances w.r.t samples $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ drawn from the global feature distribution $\varphi_j(x|y=k)$

$$d_{ij}^k = \sum_{n=1}^{M} \|h(\theta_i;\mathbf{x}_n) - h(\theta_j^*;\mathbf{x}_n)\|_1 \tag{5}$$

Once the current value d_{ij}^k for all the classes are estimated on the server side, we can get the federated model distance d_{ij} and then partition the set of clients into K clusters based on the closest cluster model, i.e., $s_i = \arg\min_j d_{ij}$.

3.3 Sampling on the server side

Despite the tractability of the probability outputs of the models, estimating the federated model distance still involves samples from the corresponding cross distribution Q_{ij}, determined by $p_i(y)$ and $\varphi_j(x|y)$, which are unknown to us in advance. To tackle that, we first approximate the label distribution $p_i(y)$ on client i using the class ratios in the local dataset $\mathcal{D}_i$. Meanwhile, the cluster models can be leveraged as an aggregation of the

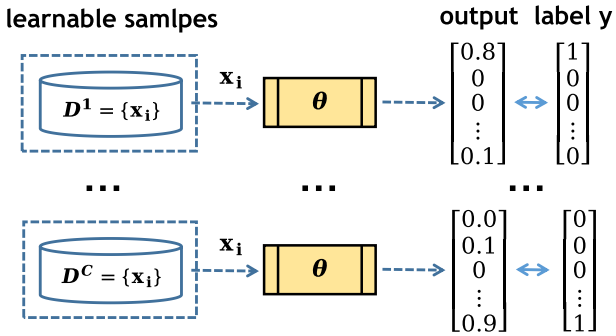

Fig. 1 Back-searching sampling

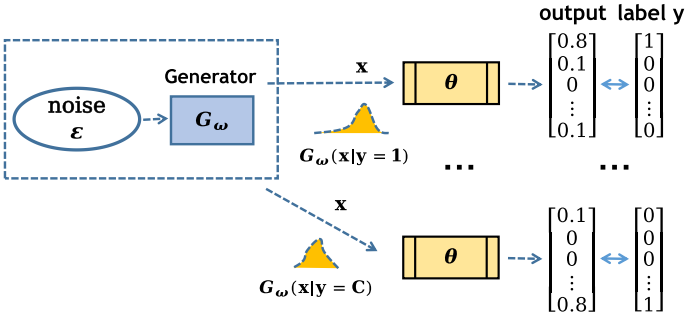

Fig. 2 Generator-based sampling

cluster information to simulate the global feature distribution $\varphi_j(x|y)$ on the server side. Next, we propose two methods to generate samples from $\varphi_j(x|y)$ approximately, including Back-searching sampling and Generator-based sampling, which are illustrated in Figs. 1 and 2 (more details in Sect. 4).

3.3.1 Back-searching sampling

In order to sample from $\varphi_j(x|y) \propto \varphi_j(y|x)\varphi_j(x)$, we initialize a pseudo dataset using Gaussian noise for every class, and optimize these samples based on the given labels and the cluster model parameters θ_j^* for a few iterations, so that each updated sample x has high probabilities for both the prior $\varphi_j(x)$ and the posterior $\varphi_j(y|x)$ w.r.t its label y. Specifically, for a classification model whose output is a probability (softmax) vector, we search for an input x which is close to the mean of the prior $\varphi_j(x)$, with the output $h(\theta_j^*, x)$ of the cluster model θ_j^* approaching $[1, 0, \cdots, 0]$ if the given label y is "1". Hence, we minimize the following objective

$$\arg\min_x f(h(\theta_j^*; x), y) + \frac{\lambda}{2}\|x - \mu(x)\|_2. \tag{6}$$

$\mu(x)$ is the mean of samples over the prior $\varphi_j(x)$ which can be estimated with some prior information determined by normalization during data preprocessing. λ is a hyperparameter

controlling the influence of the second term. The sample x which minimizes the above objective is associated with large values for both probabilities $\varphi_j(y|x)$ and $\varphi_j(x)$.

3.3.2 Generator-based sampling

Different from directly searching for samples with high probabilities in Back-searching sampling, we consider learning a conditional generator $G_\omega(x|y)$ that approximates the global feature distribution $\varphi_j(x|y)$ on the server side, where ω represents the parameters of the conditional generator. Due to privacy constraints, the server is not allowed to access real data to train the generator. To address this issue, we introduce a supervised training method motivated from Zhu et al. (2021) that uses the given one-hot label as the supervised information of the generated samples and optimize the objective below

$$\min_{\omega} J(\omega) = \mathop{\mathbb{E}}_{z\sim G_\omega(x|y)} \left[f(h(\theta_j^*;z), y) \right] + \frac{\lambda}{2}\|z - \mu(z)\|_2 \tag{7}$$

where z is the generated sample for label y and ω is the only parameter to be learned in this objective function. Note that the cluster model θ_j^* is fixed when training the conditional generator. Similar to Back-searching sampling, our goal is that the generated samples have large values of both probabilities $\varphi_j(y|x)$ and $\varphi_j(x)$. After obtaining well-trained $G_\omega(x|y)$ for every cluster j, we use them to simulate the global feature distribution $\varphi_j(x|y)$ and create a pseudo dataset for each label y to estimate the federated model distance according to Eqs. (4) and (5).

3.4 Communication-efficient framework MD-ICFL for clustered federated learning

Based on the federated model distance defined in Sect. 3.2 and the sampling methods in Sect. 3.3, for each cluster-client pair (j, i), we can estimate the class-wise model distance d_{ij}^k for $k \in [C]$ on the server side and then aggragate them using the weights $p_i(y = k)$ to obtain the final federated model distance d_{ij} as Eq. (4). For the local label distribution $p_i(y = k)$, which is unseen to the server, there are two options. Clients send the vector $\mathbf{p}_i(y) = \left[p_i(y = 1), \cdots, p_i(y = C)\right] \in \mathbb{R}^C$ to the server after the first participant communication round under weak privacy constraints where $\mathbf{p}_i(y)$ is not privacy-sensitive. Alternatively, under strong privacy constraints where the server has no access to the label distribution $\mathbf{p}_i(y)$, we can conduct a "secondary communication" that the server sends these class-wise model distance vectors $\mathbf{d}_{ij} = \left[d_{ij}^{k=1}, \cdots, d_{ij}^{k=C}\right] \in \mathbb{R}^C$ to the corresponding client i after they are estimated on the server side in each round. Client i proceeds to calculate d_{ij}'s locally and send the current cluster identity s_i back to the server. The extra communication cost introduced by this process is negligible, which is incurred by the server sending K vectors $\mathbf{d}_{ij}, j \in [K]$ additionally to each client in each round.

Algorithm 1 MD-ICFL (under weak privacy constraints)

Input: number of clusters K, number of clients N, number of local epochs E, learning rate μ, local minibatch size B.
Output: the K cluster models $\{\theta_j^{*(T)}\}$.

1: Initialize cluster models $\{\theta_j^{*(0)}\}$, cluster identities $\{s_i\}$.
2: **for** $t = 0$ to $T-1$ **do**
3: Create pseudo datasets $\{\mathbf{D}_j^k\}$ from K global feature distributions by (6) or (7) with $\{\theta_j^{*(t)}\}$.
4: $\mathbb{S}_t \leftarrow$ random subset of N clients.
5: **for** client $i \in \mathbb{S}_t$ **in parallel do**
6: $\theta_i^{(t)}, \mathbf{p}_i(y) \leftarrow$ Local-update$(\theta_{s_i}^{*(t)}, \mu, E, B)$.
7: **end for**
8: **for** client $i \in \mathbb{S}_t$ **do**
9: The server estimates d_{ij} based on (4) with $\theta_i^{(t)}$, $\{\theta_j^{*(t)}\}$, $\mathbf{p}_i(y)$, $\{\mathbf{D}_j^k\}$.
10: $s_i \leftarrow \arg\min_{j\in[K]} d_{ij}$.
11: **end for**
12: $\{\theta_j^{*(t+1)}\} \leftarrow \{\sum_{i\in\mathbb{S}_t} \frac{\mathbf{1}(s_i=j)}{|\mathbb{S}_t(j)|}\theta_i^{(t)}\}$.
13: **end for**

Formally, we propose the Model Distance based Iterative Clustered Federated Learning (MD-ICFL). The complete algorithms under weak and strong privacy constraints are described in Algorithm 1 and Algorithm 2 respectively, which both perform client clustering in an iterative way. For the weak privacy setting, we first initialize K cluster model parameters θ_j^* and cluster identities s_i for the clients randomly. In the t-th round, we simulate the process of sampling from K global feature distributions $\varphi_j(x|y)$ with only the current cluster models on the server side, thus generating K pseudo datasets $\mathbf{D}_j$, each of which contains C subsets representing C conditional distributions $\varphi_j(x|y=k)$, i.e., $\mathbf{D}_j = \{\mathbf{D}_j^k\}_{k=1}^C$ (line 3). After sampling a subset of clients $\mathbb{S}_t$, each client in $\mathbb{S}_t$ performs local updates for a few epochs and sends the updated local model $\theta_i^{(t)}$ back to the server along with the class distribution $\mathbf{p}_i(y)$ (line 4–6). We then calculate the federated model distance d_{ij} and estimate new cluster identities for each client by $s_i = \arg\min_j d_{ij}$ (line 9–10). The server aggregates the local models in $\mathbb{S}_t$ to obtain the newest cluster models θ_j^* depending on the current cluster identities s_i's (line 12). MD-ICFL for the strong privacy setting has a similar procedure except for the secondary communication (line 9–12 in Algorithm 2).

It is worth noting that generating pseudo datasets on the server side and the local updates on clients can be performed simultaneously. In addition, when the secondary communication described above is conducted under strong privacy constraints, the server cannot infer the real class distribution vector $\mathbf{p}_i(y)$ since it only receives the cluster identity s_i from client i within finite communication rounds. Specifically, we can formulate the reconstruction problem of the class distribution vector $\mathbf{p}_i(y)$ as follows

$$\begin{aligned} s_i^{(t)} &= \arg\min_{j\in[K]} \mathbf{p}_i(y)^T \mathbf{d}_{ij}^{(t)} \\ \text{s.t.} \quad \|\mathbf{p}_i(y)\|_1 &= 1, 0 \le p_i(y=k) \le 1 \\ t &= 1, 2, \cdots, t_{\max} \end{aligned} \tag{8}$$

where we use the superscript (t) in $s_i^{(t)}$ and $\mathbf{d}_{ij}^{(t)}$ to denote the t-th round. The first equation holds for all communication rounds. There is no unique solution for the target vector $\mathbf{p}_i(y)$ for all the above equations if the number of equations $t_{\max}$ is finite. Hence, the server is not able to reconstruct the real value of $\mathbf{p}_i(y)$ and this process would not lead to privacy leakage.

Algorithm 2 MD-ICFL (under strong privacy constraints)

Input: number of clusters K, number of clients N, number of local epochs E, learning rate μ, local minibatch size B.
Output: the K cluster model $\{\theta_j^{*(T)}\}$.

Initialize cluster models $\{\theta_j^{*(0)}\}$, cluster identities $\{s_i\}$.
for $t = 0$ to $T - 1$ **do**
  Create pseudo datasets $\{\mathbf{D}_j^k\}$ from K global feature distributions by (6) or (7) with $\{\theta_j^{*(t)}\}$.
  $\mathbb{S}_t \leftarrow$ random subset of N clients.
  for client $i \in \mathbb{S}_t$ **in parallel do**
    $\theta_i^{(t)} \leftarrow$ Local-update$(\theta_{s_i}^{*(t)}, \mu, E, B)$.
  end for
  The server estimates $\mathbf{d}_{ij}$ based on (5) with $\theta_i^{(t)}$, $\{\theta_j^{*(t)}\}$, $\{\mathbf{D}_j^k\}$.
  The server sends $\{\mathbf{d}_{ij} \mid j \in [K]\}$ to client $i \in \mathbb{S}_t$.
  for client $i \in \mathbb{S}_t$ in parallel **do**
    $s_i \leftarrow \arg\min_{j \in [K]} \mathbf{p}_i(x)^T \mathbf{d}_{ij}$.
  end for
  $\{\theta_j^{*(t+1)}\} \leftarrow \{\sum_{i \in \mathbb{S}_t} \frac{\mathbf{1}(s_i = j)}{|\mathbb{S}_t(j)|} \theta_i^{(t)}\}$.
end for

Algorithm 3 Local-update

Input: your algorithm's input.
Parameter: optional list of parameters.
Output: your algorithm's output.

$\mathcal{B} \leftarrow$ split dataset D into batches of size B.
for $t = 0$ to $E - 1$ **do**
  for batch $b \in \mathcal{B}$ **do**
    $\theta \leftarrow \theta - \mu \nabla f(\theta; b)$.
  end for
end for
return θ, $\mathbf{p}_i(y)$

4 Implementation of sampling methods

In this section, we present the details of our implementation for both of the sampling methods in Sect. 3. First, for Back-searching sampling, we create a pseudo dataset $\mathbf{D}_j = \{\mathbf{D}_j^k\}$ containing C subset $\mathbf{D}_j^k, k = 1, \cdots, C$ for each cluster j, one of which has $M = 30$ samples initialized with the Gaussian noise, i.e., $|\mathbf{D}_j^k| = 30$. We fix the parameters of cluster models θ_j^* and set the labels of all samples in subset $\mathbf{D}_j^k$ to k. These samples are then optimized using the Adam optimizer based on (6). We use Cross Entropy as the loss function f and set the hyper-parameter λ to 0.1. For each sample in $\mathbf{D}_j^k$, we optimize it with a learning rate of 0.1 for 100 iterations. The resulted pseudo datasets will be considered as the generated samples from $\varphi_j(x|y = k)$.

For the Generator-based sampling, we adopt a conditional generator $G_\omega(x|y)$ similar to Zhu et al. (2021); Kingma and Welling (2013), which is two-layer fully-connected network with Batch Normalization and ReLU activation function to learn the global feature distribution $\varphi_j(x|y = k)$ based on (7) for each cluster j. The input of $G_\omega(x|y)$ are the random noises ϵ drawn from a high-dimensional Gaussian distribution and the label $y \in [C]$. We set the size of the input noise to 16 for all the experiments in this paper and randomly assign the label of each input noise from 1 to C. We use the Adam optimizer with a learning rate of 0.01 and a batch-size of 100 to train the conditional generator for 1000 iterations. Finally, we get $K = 4$ conditional generators, each of which represents one global feature distribution. These conditional generators will later be used to generate the pseudo datasets $\mathbf{D}_j = \{\mathbf{D}_j^k\}$. Similar to Back-searching sampling, we draw $M = 30$ samples for each class y from each global feature distribution.

5 Experiments

In this section, we validate the effectiveness of our framework MD-ICFL in various CFL scenarios. We create label non-iid datasets following the setting in Lin et al. (2020) where the local label distribution vectors $\mathbf{p}_i(y)$ of clients are sampled from a Dirichlet distribution Dir(α). α is the hyper-parameter controlling the degree of label non-iid, i.e., class imbalance. The process of generating local data distribution $p_i(x)$ for client i is depicted in Fig. 3. $Z(s)$ is the distribution of the cluster identity which we set as a uniform distribution. For the image classification task where data is structured, we normalize each pixel of images to $[-1, 1]$ and choose $\mu = 0.5$ as the prior for all experiments.

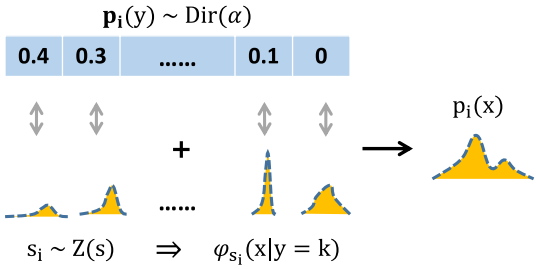

Fig. 3 Dataset generation

5.1 Experimental details

5.1.1 Dataset generation

In order to create a local dataset which is generated from a global feature distribution $\varphi_j(x|y)$ and with any possible label distribution $p_i(y)$ for client i, we perform three steps as depicted in Fig. 3: (1) Determine the cluster identity $s_i \sim Z(s)$. (2) Sample a label distribution vector $\mathbf{p}_i(y)$ from $\text{Dir}(\alpha, \mathbf{p})$. $\text{Dir}(\alpha, \mathbf{p})$ is a Dirichlet distribution with the parameters α and $\mathbf{p}$, where $\mathbf{p}$ is the global label distribution $\varphi_{s_i}(y)$ which we set to be uniform distribution over classes. (3) Construct a joint distribution $p_i(x, y)$ whose conditional (feature) distribution is $\varphi_{s_i}(x|y)$ and label distribution is $p_i(y)$ followed by drawing samples independently from the joint local distribution $p_i(x, y)$ to obtain a local dataset. We utilize a given single dataset as the global feature distribution.

We observe that the label distribution will be extremely unbalanced if α is less than or equal to 1 (where almost every client has only one or at most two classes of data when the number of class $C = 10$). Too small α makes the CFL problem pathological and leads to a contradiction with the common assumption that there is a distinct cluster structure because the underlying cluster structure may be uncertain in this case, which means there exist many other definitions regarding the K global feature distribution. In this paper we choose $\alpha = 3, 10, 100$ respectively to guarantee the unique cluster structure. Every client has three to five classes of data when $\alpha = 3$, and when $\alpha = 100$ it has access to data from all classes ($C = 10$) with an approximately uniform label distribution.

5.1.2 Label iid setting

For the Rotated MNIST and Multi-Source Digit dataset, $K = 4$ global feature distributions are used to create local datasets for $N = 48$ clients uniformly. We choose $\alpha = 100$ to generate the label distribution vector of each client in the label iid scenario. The size of each local dataset $|\mathcal{D}_i|$ is set to 1000 and the raw data in each of them is normalized with mean $\mu = 0.5$ and variance $\sigma = 0.5$. Besides, Multi-source Digit is a realistic multi-source dataset created for simulating the scenario where the variations in feature distributions are caused by style, device or personal habit. It is composed of four digit-related dataset MNIST, USPS, SVHN and SIGN (Mavi, 2020). MNIST and USPS are two handwritten digit datasets in two different styles, SVHN is a digit dataset colleted from realistic scenes, and SIGN is a sign language digit dataset. Each client draw samples from one of the four dataset as its local dataset.

We use the typical network LeNet that consists of 2 convolutional layers followed by 3 fully connected layers for both the Rotated MNIST dataset and the Multi-Source Digit dataset. For the Rotated MNIST dataset, we implement every method with $T = 30$ communication rounds for model training. Additionaly, we set the number of local epoch as $E = 2$, the learning rate as $\eta = 0.1$, and the local minibatch size as $B = 50$. Multi-Source Digit dataset follows the same settings except for $E = 3$. Full ($R = 1.0$) and partial ($R = 0.5$) client participation are adopted in our experiments respectively.

5.1.3 Label non-iid setting

The Swapped Rotated CIFAR10 dataset is constructed based on the CIFAR10 dataset with the same rotation operation used in the Rotated MNIST dataset. Besides, we notice that the classes in CIFAR10 are indeed not highly related to specific angles, unlike the case in digits. To increase the dissimilarity between different feature distributions, we swap the labels of two given classes within each cluster. For example, we modify the labels of data points labeled as "1" to "2" and vice versa in the first cluster. As a result, each cluster has 4 (40%) classes different from another cluster in concept. This operation was first adopted in Sattler and Müller (2020) to simulate an incongruent clustering structure for the "data incongruence" issue, which can be considered as a kind of concept drift caused by personalities. We create $N = 48$ clients with $K = 4$. The size of each local dataset is set as $|\mathcal{D}_i| = 5000$.

We use two commonly used deep learning models ResNet18 and MobileNetV2 for all experiments on this dataset. MobileNetV2 is a light neural network designed for mobile devices and thus suitable for FL scenarios. For every method, we execute it in $T = 60$ communication rounds, with the number of local epoch $E = 1$, the learning rate $\eta = 0.1$, and the local minibatch size $B = 50$.

5.2 Baselines

In all experiments, we validate and compare our framework MD-ICFL with four baselines: (1) **Global model**: a single global model is learned that can fit K global feature distributions simultaneously. (2) **Local model**: each client trains a personalized model only on its local dataset. (3) **IFCA** (Ghosh et al., 2020): the Iterative Federated Clustering Algorithm which performs client clustering iteratively based on the values of loss functions. (4) **FedFomo** (Zhang et al., 2020): the personalized federated learning algorithm achieving the SOTA performance in most of the data heterogeneity problems. (5) **ClusteredFL** (Sattler and Müller, 2020): the client populations are grouped into clusters with jointly trainable data distributions. We denote our framework MD-ICFL with back-searching sampling and generator-based sampling as **MD-ICFL-1** and **MD-ICFL-2** respectively.

We evaluate the performance of all methods via the **averaged accuracy** on test datasets over all clients after the same epochs. For CFL-related algorithms, i.e., IFCA and our framework MD-ICFL, averaged Adjusted Rand Index (**ARI**) will be computed based on

Table 1 Averaged test accuracy and ARI (for IFCA and MD-IFCL-1 &2) on swapped rotated CIFAR10 ($\alpha = 3$)

Model	ResNet18	MobileNetV2
Global model	61.68	59.63
Local model	37.48	19.52
FedFomo	44.44	28.83
ClusteredFL	12.45	10.27
IFCA	65.69 (0.51)	50.84 (0.23)
MD-ICFL-1	**74.85** (0.80)	75.2 (0.89)
MD-ICFL-2	74.41 (**0.84**)	**76.32** (**0.94**)

Our proposed methods and the best score for each evaluation metric are bolded

Table 2 Averaged test accuracy and ARI (for IFCA and MD-IFCL-1 &2) on swapped rotated CIFAR10 ($\alpha = 10$)

Model	ResNet18	MobileNetV2
Global model	67.29	64.56
Local model	57.73	26.99
FedFomo	61.89	59.61
ClusteredFL	23.60	18.07
IFCA	75.62(0.57)	57.59 (0.16)
MD-ICFL-1	80.21 (0.93)	77.88 (0.83)
MD-ICFL-2	**80.96** (**0.99**)	**79.46** (**0.95**)

Our proposed methods and the best score for each evaluation metric are bolded

Table 3 Averaged test accuracy and ARI (for IFCA and MD-IFCL-1 &2) on swapped rotated CIFAR10 ($\alpha = 100$)

Model	ResNet18	MobileNetV2
Global model	69.20	65.48
Local model	63.06	35.03
FedFomo	72.42	71.72
ClusteredFL	45.40	44.37
IFCA	72.7 (0.24)	58.46 (0)
MD-ICFL-1	81.67 (0.98)	80.93 (0.94)
MD-ICFL-2	**82.17** (**1.00**)	**81.52**(**0.97**)

Our proposed methods and the best score for each evaluation metric are bolded

the ground-truth cluster identities to evaluate the results of client clustering. ARI takes values in $[-1, 1]$ and a larger ARI value indicates better performance of clustering.

5.3 Results on label non-iid data

In order to demonstrate the superiority of our framework in the label non-iid scenarios, we first validate MD-ICFL and the baselines on datasets with different label non-iid levels controlled by the hyper-parameter α. To simulate the clustered federated learning scenario where the local datasets on clients are generated from K global feature distributions $\varphi_j(x|y)$, a new dataset named Swapped Rotated CIFAR10 ($K = 4$) is introduced here. We construct the dataset based on the method used in Ghosh et al. (2020) where the CIFAR10 dataset is augmented by applying 0, 90, 180 and 270 degrees of rotation to each image. The augmented data of each angle can be viewed as an independent global feature distribution. Consider that some classes in CIAFR10 are not highly related to specific angles, we utilize an additional label swapping operation introduced in Sattler and Müller (2020) to increase the dissimilarities between different clusters. We choose $\alpha = 3, 10, 100$ respectively corresponding to three label non-iid levels. Two commonly-used deep learning models ResNet18 and MobileNetV2 are trained on the Swapped Rotated CIFAR10 dataset with full participation ($R = 1.0$). The experimental results with $\alpha = 3, 10, 100$ are shown in Tables 1, 2 and 3 respectively.

Results above show that our framework consistently outperforms other baselines with large margins in different label non-iid scenarios. As the degree of label non-iid increases

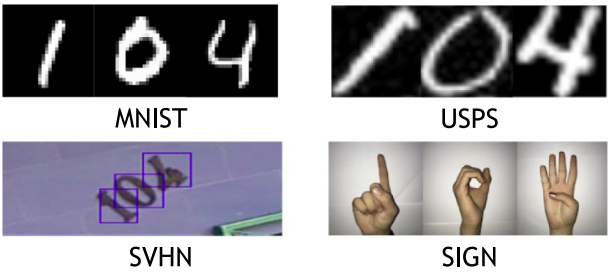

Fig. 4 Multi-source digits

(α varies from 100 to 3), our framework shows more significant advantages than other compared methods. The ARI values of our framework in three tables are all larger than 0.8, which means a desirable clustering performance can be achieved even if the label distributions across clients are different. This can be attributed to the new model distance that can explicitly measure the class-wise dissimilarities. In contrast, the hierarchical clustering method ClusteredFL fails in the label non-iid settings as ClusteredFL conducts clustering based on the gradients of local loss functions which are defined by class-imbalanced datasets. Meanwhile, IFCA can hardly identify the correct cluster structure and FedFomo performs worse since the testing distribution is quite different from the training distributions on each client,

Table 4 Averaged test accuracy and ARI (for IFCA and MD-IFCL-1 &2) on rotated MNIST ($\alpha = 100$)

	R=0.5	R=1.0
Global model	91.04	91.18
Local model	–	93.69
FedFomo	94.56	94.60
ClusteredFL	92.34	92.45
IFCA	**97.47** (0.88)	97.17 (0.76)
MD-ICFL-1	97.39 (**0.94**)	**97.28** (**0.95**)
MD-ICFL-2	97.44 (0.93)	97.20 (0.90)

Our proposed methods and the best score for each evaluation metric are bolded

Table 5 Averaged test accuracy and ARI (for IFCA and MD-IFCL-1 &2) on multi-source digits ($\alpha = 100$)

	R=0.5	R=1.0
Global model	86.24	86.81
Local model	–	85.01
FedFomo	86.06	86.43
ClusteredFL	73.29	73.41
IFCA	**91.62** (0.7)	**92.14** (0.7)
MD-ICFL-1	90.08 (0.69)	90.80 (**0.79**)
MD-ICFL-2	90.46 (**0.73**)	90.19 (0.65)

Our proposed methods and the best score for each evaluation metric are bolded

Table 6 Averaged test accuracy on ratated MNIST($\alpha = 100$) with different K

	MD-ICFL-1	MD-ICFL-2
K = 1	92.30	91.56
K = 2	95.68	94.19
K = 3	97.01	96.94
K = 4	97.28	97.20
K = 5	97.25	97.27
K = 6	97.18	97.16

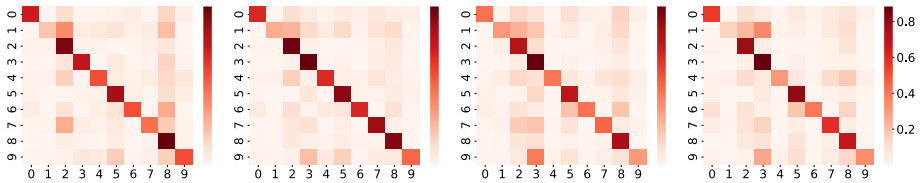

Fig. 5 The outputs of new models on pseudo dataset from Back-searching sampling

especially when $\alpha = 3$, i.e., the highest level of label non-iid. Hence, our framework is an effective iterative CFL method with less communication cost in the label non-iid scenario.

5.4 Results on label iid data

We further validate our framework with baselines on label iid datasets. The vectors for the label distributions of clients in this scenario are generated by setting $\alpha = 100$. We construct two datasets: (1) Rotated MNIST: the dataset introduced in Ghosh et al. (2020) where the MNIST dataset is augmented with the same rotation operation as the Swapped Rotated CIFAR10 dataset. (2) Multi-source Digits: four digit-related datasets including MNIST, USPS, SVHN and SIGN (Mavi, 2020) representing four different global feature distributions (Fig. 4). Both datasets are trained with the classical LeNet. Experimental results are reported in Tables 4 and 5 where the ARI values are listed in the parenthesises following the test accuracies for the CFL-related algorithms.

The results show that MD-ICFL achieves comparable test accuracy and better clustering performance on both datasets. The larger averaged ARI values in most cases indicate that our framework is more robust to the initial cluster models and thus not prone to model degradation where the number of clusters becomes less than K during training. We notice that in Multi-source Digits, MNIST is so similar to USPS that IFCA tends to quickly converge to only 3 clusters and leads to slightly better test accuracy, while our method could still find the correct cluster structure as implied by the larger ARI value in MD-ICFL-1 when $R = 1.0$. Recall that our framework requires only about $1/K$ communication cost for clients when downloading models in each round compared to IFCA. Besides, other baselines including FedFomo and ClusteredFL either do not consider the cluster structure of the clients or work in a multi-task learning manner, leading to lower averaged test accuracy.

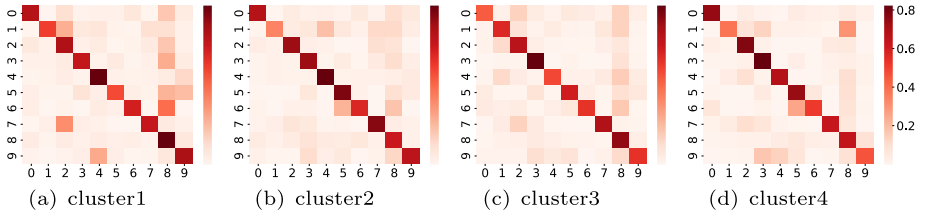

Fig. 6 The outputs of new models on pseudo dataset from Generator-based sampling

5.5 Effects of K

To investigate the effects of K, we vary the value of K while fixing the real number of clusters to 4 and show the accuracy on Rotated MNIST in Table 6. Although the performance of the model drops significantly when K is too small, it has little impact when the K is larger than the real number of clusters. Therefore, in real applications, we can set the value of K slightly larger to ensure the performance of the model.

5.6 Visualization of Pseudo datasets

In order to demonstrate the effectiveness of the two sampling methods in Sect. 3, in this section we show that the generated pseudo samples are some meaningful and cluster-biased points in the sample space rather than random noises. Specifically, we claim that generated pseudo datasets $\{\mathbf{D}_j^k\}$ have high probabilities or confidence for its corresponding posterior distribution $\varphi_j(y|x)$ indeed. We first perform our framework MD-ICFL-1 &2 for the task on the Rotated MNIST dataset (introduced in Sect. 5) and get the K optimized cluster models in the case where all clients are clustered correctly, based on which we generate pseudo datasets $\{\mathbf{D}_j^k\}$ with more samples for each k and j. We test these pseudo datasets on $K = 4$ new models θ_j^{new}, each of which is directly trained with a single global feature distribution, i.e, the complete MNITS dataset with the degree of rotation 0, 90, 180, or 270. Considering the the random nature of model initialization, we train these new models 5 times by initialing them using different random seeds. The averaged probability outputs of the pseudo samples in $\mathbf{D}_j^k$ of the corresponding model θ_j^{new} over 5 random seeds are visualized in Figs. 5 and 6. Each sub-figure presents the probability outputs over 10 classes (row) w.r.t the pseudo samples drawn from $\varphi_j(x|y), y = 0, 1, \cdots, 9$ (column).

We observe that the high values of probabilities are almost located on the diagonal of the matrix. This means that these samples generated from $\varphi_j(x|y)$ have high probabilities of the posterior $\varphi_j(y|x)$ in general. Meanwhile, we also notice that the generated pseudo samples are visually indistinguishable (see Fig. 7). Therefore, our sampling methods

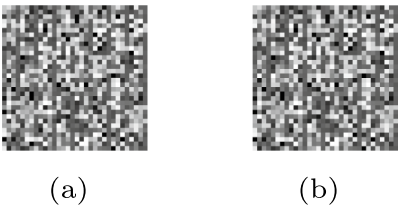

Fig. 7 The examples of generated pseudo samples. **a** Back-searching sampling. **b** Generator-based sampling

can be regarded as generating some points with high confidence w.r.t the global feature distribution $\varphi_j(x|y)$, which are enough to be used for representing the dissimilarity among distributions while not leaking the private raw data on participant clients.

6 Conclusion and discussion

In this work, we introduce an iterative framework MD-ICFL for Clustered Federated Learning from the perspective of model distance. Our method can estimate the model distances between cluster models and client models on the server side and thus requires about $1/K$ downloading communication cost compared to the existing iterative CFL method IFCA. Experimental results show the effectiveness of our framework under several CFL scenarios, especially the label non-iid settings. Moreover, in order to validate the sampling methods proposed in the paper, we show that these generated pseudo data instances indeed correspond to samples with high confidence.

Author contribution MZ is first author. LX is corresponding author. TZ, YC CB, HC and DJ are co-authors.

Funding This research was supported by the National Natural Science Foundation of China (Grant no. 62276245), and Anhui Provincial Natural Science Foundation (Grant no. 2008085J31).

Data availibility The datasets used in the study are publicly available from their corresponding authors.

Code availability The code for this study are not publicly available until the paper is published.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate The authors agree to participate.

Consent for publication The authors agree to the publication of the data and images in this paper.

References

Chen, H., Tino, P., Rodan, A., et al. (2013). Learning in the model space for cognitive fault diagnosis. *IEEE Transactions on Neural Networks and Learning Systems, 25*(1), 124–136.

Fu, Y., Liu, X., & Tang, S., et.al. (2021) Cic-fl: Enabling class imbalance-aware clustered federated learning over shifted distributions. In International conference on database systems for advanced applications (pp. 37–52). Springer.

Ghosh, A., Hong, J., & Yin, D. (2019). Robust federated learning in a heterogeneous environment. arXiv preprint arXiv:1906.06629

Ghosh, A., Chung, J., & Yin, D., et al. (2020). An efficient framework for clustered federated learning. arXiv preprint arXiv:2006.04088

Huang, Y., Chu, L., Zhou, Z., et al. (2021). Personalized cross-silo federated learning on non-iid data. In Proceedings of the AAAI conference on artificial intelligence (pp. 7865–7873).

Kairouz, P., McMahan, H. B., Avent, B., et al. (2019). Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977

Karimireddy, S. P., Kale, S., & Mohri, M. (2020). Scaffold: Stochastic controlled averaging for federated learning. In ICML, PMLR (pp. 5132–5143).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114

Li, T., & Sahu, A. K. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine, 37*(3), 50–60.

Li, X., Huang, K., & Yang, W., et.al. (2019). On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189

Lin, T., Kong, L., Stich, S. U., et al. (2020). Ensemble distillation for robust model fusion in federated learning. *NIPS, 33*, 2351–2363.

Liu, B., Guo, Y., & Chen, X. (2021). Pfa: Privacy-preserving federated adaptation for effective model personalization. In Proceedings of the web conference (vol. 2021, pp. 923–934).

Mavi, A. (2020) A new dataset and proposed convolutional neural network architecture for classification of american sign language digits. arXiv preprint arXiv:2011.08927

McMahan, B., Moore, E., & Ramage, D. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, PMLR (pp. 1273–1282).

Sattler, F., & Müller, K. R. (2020). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems, 32*(8), 3710–3722.

Smith, V., Chiang, C. K., Sanjabi, M., et al. (2017). Federated multi-task learning. arXiv preprint arXiv:1705.10467

Wang, H., Yurochkin, M. (2020). Federated learning with matched averaging. arXiv preprint arXiv:2002.06440

Wang, K., Mathews, R., Kiddon, C., et al. (2019). Federated evaluation of on-device personalization. arXiv preprint arXiv:1910.10252

Zhang, M., Sapra, K., Fidler, S., et.al. (2020). Personalized federated learning with first order model optimization. arXiv preprint arXiv:2012.08565

Zhu, Z., Hong, J., Zhou, J. (2021). Data-free knowledge distillation for heterogeneous federated learning. In ICML, PMLR (pp. 12878–12889).

Authors and Affiliations

Mao Zhang[1,3] · Tie Zhang[1,3] · Yifei Cheng[2,3] · Changcun Bao[4] · Haoyu Cao[4] · Deqiang Jiang[4] · Linli Xu[1,3]

✉ Linli Xu
linlixu@ustc.edu.cn

Mao Zhang
zmyyy@mail.ustc.edu.cn

Tie Zhang
tiezhang@mail.ustc.edu.cn

Yifei Cheng
chengyif@mail.ustc.edu.cn

Changcun Bao
changcunbao@tencent.com

Haoyu Cao
rechycao@tencent.com

Deqiang Jiang
dqiangjiang@tencent.com

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

[2] School of Data Science, University of Science and Technology of China, Hefei, China

[3] State Key Laboratory of Cognitive Intelligence, Hefei, China

[4] Tencent YouTu Lab, Hefei, China