Efficient Distribution Similarity Identification in Clustered Federated Learning via Principal Angles Between Client Data Subspaces

Saeed Vahidian\textsuperscript{1}, Mahdi Morafah\textsuperscript{1}, Weijia Wang\textsuperscript{1}, Vyacheslav Kungurtsev\textsuperscript{2}, Chen Chen\textsuperscript{3}, Mubarak Shah\textsuperscript{3}, Bill Lin\textsuperscript{1}

\textsuperscript{1}UC San Diego
\textsuperscript{2}Czech Technical University
\textsuperscript{3}UCF

\{svahidia, mmorafah, wweijia, billin\}@ucsd.com, kunguvya@fel.cvut.cz, \{chen.chen, shah\}@crcv.ucf.edu

Abstract

Clustered federated learning (FL) has been shown to produce promising results by grouping clients into clusters. This is especially effective in scenarios where separate groups of clients have significant differences in the distributions of their local data. Existing clustered FL algorithms are essentially trying to group together clients with similar distributions so that clients in the same cluster can leverage each other’s data to better perform federated learning. However, prior clustered FL algorithms attempt to learn these distribution similarities indirectly during training, which can be quite time consuming as many rounds of federated learning may be required until the formation of clusters is stabilized. In this paper, we propose a new approach to federated learning that directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces. Each client applies a truncated singular value decomposition (SVD) step on its local data in a single-shot manner to derive a small set of principal vectors, which provides a signature that succinctly captures the main characteristics of the underlying distribution. This small set of principal vectors is provided to the server so that the server can directly identify distribution similarities among the clients to form clusters. This is achieved by comparing the similarities of the principal angles between the client data subspaces spanned by those principal vectors. The approach provides a simple, yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews. Our clustered FL approach also enables convergence guarantees for non-convex objectives.

Introduction

Federated Learning (FL) (McMahan and Ramage 2017) enables a set of clients to collaboratively learn a shared prediction model without sharing their local data. Some FL approaches aim to train a common global model for all clients (McMahan et al. 2017; Li et al. 2020; Wang et al. 2020; Karimireddy et al. 2020; Mendieta et al. 2022). However, in many FL applications where there may be data heterogeneity among clients, a single relevant global model may not exist. Alternatively, personalized FL approaches have been studied. One approach is to first train a global model and then allow each client to fine-tune it via a few rounds of stochastic gradient descent (SGD) (Fallah, Mokhtari, and Ozdaglar 2020; Vahidian et al. 2022; Liang et al. 2020a). Another approach is for each client to jointly train a global model as well as a local model, and then interpolate them to derive a personalized model (Deng, Kamani, and Mahdavi 2020; Mansour et al. 2020). In the former case, the approach often fails to derive a model that generalizes well to the local distributions of each client. In the latter case, when local distributions and the average distribution are far apart, the approach often degenerates to every client learning only on its own local data. Recently, clustered FL (Ghosh et al. 2020; Sattler, Müller, and Samek 2021; Morafah et al. 2022; Mansour et al. 2020) has been proposed to allow the grouping of clients into clusters so that clients belonging to the same cluster can share the same optimal model. Clustered FL has been shown to produce significantly better results, especially when separate groups of clients have significant differences in the distributions of their local data. This possibly due to distinct learning tasks or the mixture of distributions of the local data considered, not necessarily limited to simpler forms of data heterogeneity such as label skews from otherwise the same dataset.

Essentially, what prior clustered FL algorithms are trying to do is to group together clients with similar distributions so that clients in the same cluster can leverage each other’s data to perform federated learning more effectively. Previous clustered FL algorithms attempt to learn these distribution similarities indirectly when clients learn the cluster to which they should belong as well as the cluster model during training. For example, the clustered FL approach presented in (Sattler, Müller, and Samek 2021) alternately estimates the cluster identities of clients and optimizes the cluster model parameters via SGD.

Unfortunately, the prior clustered FL approaches have the following challenges which in turn limits their applicability in real-world problems. 1) Since previous clustered FL training algorithms start with randomly initialized cluster models that are inherently noisy, the overall training process can be quite time consuming as many rounds of federated learning may be required until the formation of clusters is stabilized. 2) Approaches like IFCA (Ghosh et al. 2020) assumes a
which require time consuming iterative learning of the client parameters, which each client would be best trained on just its own local data (i.e., each client becomes its cluster).

In each iteration, all cluster models have to be downloaded by the active clients in that round, which can be very costly in communications. 3) Both of the approaches i.e., those that train a common global model for all clients and personalized approaches including IFCA lack the flexibility to trade off between personalization and globalization. The above-mentioned drawbacks of the prior works, naturally lead to the following important question. How a server can realize clustered FL efficiently by grouping the clients into clusters in a one-shot manner without requiring the number of clusters to be known apriori, but with substantially less communication cost? In this work, we propose a novel algorithm, Principal Angles analysis for Clustered Federated Learning (PACFL), to address the above-mentioned challenges of clustered FL.

Our contributions. We propose a new algorithm, PACFL, for federated learning that directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces. Each client wishing to join the federation applies a truncated SVD step on its local data in a one-shot manner to derive a small set of principal vectors, which form the principal bases of the underlying data. These principal bases provide a signature that succinctly captures the main characteristics of the underlying distribution. The client then provides this small set of principal vectors to the server so that the server can directly identify distribution similarities among the clients to form clusters. The privacy of data is preserved since no client data is ever sent to the server but a few (2-5) principal vectors out of ≈ 500. Thus, the clients data cannot be reconstructed from those (2-5) number of left singular vectors. However, in privacy sensitive setups to provide extra protection and prevent any information leakage from clients to server, mechanisms like the ones presented in (Bonawitz et al. 2017), or encryption mechanism or differential privacy method that achieves this end can be employed.

On the server side, it efficiently identifies distribution similarities among clients by comparing the principal angles between the client data subspaces spanned by the provided principal vectors – the greater the difference in data heterogeneity between two clients, the more orthogonal their subspaces. Unlike prior clustered FL approaches, which require time consuming iterative learning of the clusters and substantial communication costs, our approach provides a simple yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews. Clients can immediately collaborate with other clients in the same cluster from the get go.

Our novel PACFL approach has the flexibility to trade off between personalization and globalization. PACFL can naturally span the spectrum of identifying IID data distribution scenarios in which all clients should share training within only 1 cluster, to the other end of the spectrum where clients have extremely Non-IID data distributions in which each client would be best trained on just its own local data (i.e., each client becomes its cluster).

Our framework also naturally provides an elegant approach to handle newcomer clients unseen at training time by matching them with a cluster model that the client can further personalized with local training. Realistically, new clients may arrive to the federation after the distributed training procedure. In our framework, the newcomer client simply provides its principal vectors to the server, and the server identifies via angle similarity analysis which existing cluster model would be most suitable, or the server can inform the client that it should train on its own local data to form a new cluster if the client’s data distribution is not sufficiently similar to the distributions of the existing clusters. On the other hand, it is generally unclear how prior personalized or clustered FL algorithms can be extended to provide newcomer clients with similar capabilities.

Finally, we provide a convergence analysis of PACFL in the supplementary material.

Clustered Federated Learning

Preliminaries

Principal angles between two subspaces. Let \( U = \text{span}\{u_1, ..., u_p\} \) and \( W = \text{span}\{w_1, ..., w_q\} \) be \( p \) and \( q \)-dimensional subspaces of \( \mathbb{R}^n \) where \( \{u_1, ..., u_p\} \) and \( \{w_1, ..., w_q\} \) are orthonormal, with \( 1 \leq p \leq q \). There exists a sequence of \( p \) angles \( 0 \leq \Theta_1 \leq \Theta_2 \leq ... \leq \Theta_p \leq \pi/2 \) called the principal angles, which are defined as

\[
\Theta(U,W) = \min_{u \in U, w \in W} \arccos \left( \frac{|u^T w|}{||u|| ||w||} \right)
\]

where \( ||.|| \) is the induced norm. The smallest principal angle is \( \Theta_1(u_1,w_1) \) with vectors \( u_1 \) and \( w_1 \) being the corresponding principal vectors. The principal angle distance is a metric for measuring the distance between subspaces (Jain, Netrapalli, and Sanghavi 2013). Additional background is presented in the supplementary material.
Table 1: This table shows how distribution similarities between datasets can be accurately estimated by a proximity matrix of principal angles. Entries are $x(y)$, where $x$ and $y$ are respectively the smallest principal angle and summation over the principal angles between two datasets. $p$ in $U_p$ is 2.

| Dataset | CIFAR-10 | SVHN | FMNIST | USPS |
|---------|----------|------|--------|------|
| CIFAR-10 | 0 (0) | 6.13 (12.3) | 45.79 (91.6) | 66.26 (132.5) |
| SVHN | 6.13 (12.3) | 0 (0) | 43.42 (86.8) | 64.86 (129.7) |
| FMNIST | 45.79 (91.6) | 43.42 (86.8) | 0 (0) | 43.36 (86.7) |
| USPS | 66.26 (132.5) | 64.86 (129.7) | 43.36 (86.7) | 0 (0) |

How Principal Angles Can Capture the Similarity Between Data/Features

The cosine similarity is a distance metric between vectors that is known to be more tractable and interpretable than the alternatives while exhibiting high precision clustering properties, see, e.g. (Qian et al. 2004). The idea is to note that for any two vectors $x$ and $y$, by the dot product calculation $x \cdot y = \|x\|\|y\| \cos \theta$, we see that inverting the operation to solve for $\theta$ yields the angle between two vectors as emanating from the origin, which is a scale-invariant indication of their alignment. It presents a natural geometric understanding of the proportional volume of the embedded space that lies between the two vectors. Finally, by choosing to cluster using the data rather than the model, the variance of each SGD sample declines resulting in smoother training. By contrast, clustering by model parameters has the effect of increasing the bias of the clients’ models to be closer to each other.

In order to obtain a computationally tractable small set of vectors to represent the data features, we propose to apply truncated SVD on each dataset. We take a small set of principal vectors, which form the principal bases of the underlying data distribution. Truncated SVD is known to yield a good quality balance between computational expense and representative quality of representative subspace methods (Talwalkar et al. 2013). Assume there are $K$ number of datasets. We propose to apply truncated SVD (detailed in the supplementary material) on these data matrices, $D_k, k = 1, ..., K$, whose columns are the input features of each dataset. Further, let $U^p_k = [u_1, u_2, ..., u_p], (p \ll \text{rank}(D_k))$ be the $p$ most significant left singular vectors for dataset $k$. We constitute the proximity matrix $A$ as in Eq. 2 whose entries are the smallest principle angle between the pairs of $U^p_k$ or as in Eq. 3 whose entries are the summation over the angle in between of the corresponding $u$ vectors (in identical order) in each pair within $U^p_k$, where $\text{tr}(.)$ is the trace operator, and

$$A_{i,j} = \Theta_1 \left( U^i_p, U^j_p \right), \ i, j = 1, ..., K \quad (2)$$

$$A_{i,j} = \text{tr} \left( \arccos \left( U^i_p \cdot U^j_p \right) \right), \ i, j = 1, ..., K \quad (3)$$

The smaller the entry of $A_{i,j}$ is, the more similar datasets $i$ and $j$ are. Before we proceed further, through some experiments on benchmark datasets, we highlight how the proposed method perfectly distinguishes different datasets based on their hidden data distribution by inspecting the angle between their data subspaces spanned by their first $p$ left singular vectors. For a visual illustration of the result, we refer to Fig. 1. As can be seen the principal angle between the subspaces of CIFAR-10 and SVHN is smaller than that of CIFAR-10 and USPS. Table 1 shows the exact principal angles between every pairs of these datasets’ subspaces. The entries of this table is presented as $x(y)$, where $x$ is the smallest principal angle between two datasets obtained from Eq. 2, and $y$ is the summation over the principal angles between two datasets obtained from Eq. 3. Table 1 reveals that the similarity and dissimilarity of the four different datasets have been accurately captured by the proposed method. We will provide more examples in the supplementary material and will show that the similarity/dissimilarity being captured by the proposed method is consistent with well-known distance measures between two distributions including Bhattacharyya Distance (BD), Maximum Mean Discrepancy (MMD) (Gretton et al. 2012), and Kullback–Leibler (KL) distance (Hershey and Olsen 2007).

Overview of PACFL

In this section, we begin by presenting our PACFL framework. The proposed approach, PACFL, is described in Algorithm 1. We first turn our attention to clustering clients data in a federated network. The proposed method is one-shot clustering and can be used as a simple pre-processing stage to characterize personalized federated learning to achieve superior performance relative to the recent iterative approach for clustered FL proposed in (Ghosh et al. 2020). Before federation, each available client, $k$, performs truncated SVD on its own data matrix, $D_k$, and sends the $p$ most significant left singular vectors $U_{ij}$, as their data signature to the central server. Next, the server obtains the proximity matrix $A$ as in Eq. 2 or Eq. 3 where $K = |S_k|$, and $S_k$ is the set of available clients. When the number of clusters is unknown, for forming disjoint clusters, the server can employ agglomerative hierarchical clustering (HC) (Day and Edelsbrunner 1984) on the proximity matrix $A$. For more details on HC, please see the supplementary material. Hence, the cluster ID of clients is determined.

For training, the algorithm starts with a single initial model parameters $\theta^0$. In the first iteration of PACFL a random subset of available clients $S_k \subseteq [N], |S_k| = n$ is selected by the server and the server broadcasts $\theta^0$ to all clients. The clients start training on their local data and perform some steps of stochastic gradient descent (SGD) updates, and accurately. However, theoretically and rigorously speaking, when the number of the principal vectors, $p$, is bigger than 1, it can happen that one of the principal vectors of client $k$ yields a small angle with its corresponding one for client $k'$ while the other principal vectors of client $k$ yield big angle with their corresponding ones for client $k'$. With that in mind, Eq. 3 is a more rigorous measure and it always truly captures the similarity between the client data subspaces.

Considering a client owns $M$ data samples, each including $N$ features, we assumed that the $M$ data samples are organized as the columns of a matrix $D_k \in R^{N \times M}$.  

10045
Algorithm 1: PACFL

1: **Require:** Number of available clients $N$, sampling rate $R \in (0, 1]$, clustering threshold $\beta$
2: **Server:** Initialize the server model with $\theta^0$.
3: for each round $t = 1, 2, \ldots$ do
4:    $m_t \leftarrow \max(R \cdot N, 1)$
5:    $S_t \leftarrow \{k_1, \ldots, k_n\}$ % set of $n$ available clients %
6:    for each client $k \in S_t$ in parallel do
7:        if $t = 1$ % It is done in one-shot % then
8:            client $k$ sends $U^k$ to the server
9:            $\mathbf{U} = [U^1_k, \ldots, U^B_k]$ % generated by HC %
10:        else if $k$ is a new arriving client then
11:            client $k$ sends its cluster ID to the server
12:            $\theta^0 \leftarrow \theta^0_k$ % initializing all clusters with $\theta^0_k$ %
13:            else if $k$ is an existing client then
14:                client $k$ sends $U^k$ to the server
15:            $\mathbf{A}, \mathbf{U} = \text{PME}(\mathbf{A}, \mathbf{U}, \mathbf{U}^x) \% \text{Alg. 2.2} \%
16:            $\{C_1, \ldots, C_Z\} \leftarrow \text{HC}(\mathbf{A}, \beta)$ % Update the clusters, determine the cluster ID of new client %
17:                client $k$ receives the corresponding cluster model $\theta^0_z$ from the server
18:                else
19:                    client $k$ sends its cluster ID to the server and receives the corresponding cluster model $\theta^0_z$ from the server
20:        end if
21:        $\theta^0_{k,z} \leftarrow \text{ClientUpdate}(k; \theta^0_{z,k})$; by SGD training
22:    end for
23:    for $z = 1 : Z$ % $Z$ is the number of formed clusters % do
24:        $\theta^0_{z+1} = \sum_{k \in C_z} \frac{|D_k| \theta^0_{k,z}}{\sum_{k \in C_z} |D_k|}$ model averaging for each cluster %
25:    end for
26: end for

get the updated model. The clients will only need to send their cluster membership ID and model parameters back to the central server. After receiving the model and cluster ID memberships from all the participating clients, the server then collects all the parameter updates from clients whose cluster ID are the same and conducts model averaging within each cluster. It is noteworthy that in Algorithm 1, $\beta$ stands for the Euclidean distance between two clusters and is a parameter in HC.

Desirable properties of PACFL. Unlike prior work on clustered federated learning (Ghosh et al. 2020; Sattler, Müller, and Samek 2021), PACFL has much greater flexibility in the following sense. First, from a **practical** perspective, one of the desirable properties of PACFL is that it can handle partial participation of clients. In addition, PACFL does not require to know in advance whether certain clients are available for participation in the federation. Clients can join and leave the network abruptly. In our proposed approach, the new clients that join the federation just need to send their data signature to the server and the server can easily determine the cluster IDs of the new clients by constituting a new proximity matrix without altering the cluster IDs of the other clients. In PACFL, the prior information about the availability of certain clients is not required.

Second, PACFL can form the best fitting number of clusters, if a fixed number of clusters is not specified. However, in IFCA (Ghosh et al. 2020), the number of clusters has to be known apriori. Third, one-shot client clustering can be placed by PACFL for the available clients before the federation and the prior information about the availability and the number of certain clients is not required. In contrast, IFCA constructs the clusters iteratively by alternating between client identification estimation and loss function minimization which is costly in communication.

Fourth, PACFL does not add significant additional computational overhead to the FedAvg baseline algorithm as it only requires running one-shot HC clustering before training. With that in mind, the computational complexity of the PACFL algorithm is the same as that of FedAvg plus the computational complexity of HC in one-shot ($(O(N^2))$ where $N$ is the total number of clients).

Fifth, in case of either a certain and a fixed number of clients are not available at the initial stage or clients join and leave the network abruptly, clustering by PACFL can easily be applied in a few stages as outlined in Algorithm 2. Each new clients that become available for the federation, sends the signature of its data to the server and the server aggregates the signature of existing clients and the new ones as in $U_{\text{extended}}$ (Algorithm 2). Next, the server obtains the proximity matrix $\mathbf{A}$ as in Eq. 3 or Eq. 4 where all the new clients included. By **keeping the same distance threshold** as before, the cluster ID of the new clients are determined without changing the cluster ID of the old clients.

 Sixth, we should note that we tried some other clustering methods including graph clustering methods (Hallac, Leskovec, and Boyd 2015; Sarcheshmehpour, Leinonen, and Jung 2021) for PACFL and we noticed that the clustering
we compare PACFL against the following set of baselines. Works we can obtain guarantees for nonconvex objectives, for baselines that train a single global model across all clients, we compare with FedAvg (McMahan et al. 2017), to evaluate our method. For all experiments, we assume 100 clients are available and 10% of them are sampled randomly at each round. Unless stated otherwise, throughout the experiments, the number of communication rounds is 200 and each client performs 10 locals epochs with batch size of 10 and local optimizer is SGD. We let $p \in U_p$ be 3-5. Please refer to the supplementary material for more details about the experimental setup.

**Overall Performance**

We compare PACFL with all the mentioned SOTA baselines for two different widely used Non-IID settings, i.e. Non-IID label skew, and Non-IID Dirichlet label skew (Li et al. 2021a). We present the results of Non-IID label skew in the main paper and that of the Non-IID Dirichlet label skew in the supplementary material. We report the mean and standard deviation for the average of final local test accuracy across all clients over 3 runs.

**Non-IID Label Skew.** In this setting, we first randomly assign $\varphi \%$ of the total available labels of a dataset to each client and then randomly distribute the samples of each label amongst clients own those labels as in (Li et al. 2021b). In our experiments we use Non-IID label skew 20%, and 30%, i.e. $\varphi = \{20, 30\}$% respectively. Table 2 shows the results for Non-IID label skew 20%. We report the results of Non-IID label skew 30% in the supplementary material. As can be seen, global FL baselines, i.e. FedAvg, FedProx, FedNova, and SCAFFOLD perform very poorly. That’s due to weight divergence and model drift issues under heterogeneous setting (Zhao et al. 2018).

We can observe from Table 2 that PACFL consistently outperforms all SOTA on all datasets. In particular, focusing on CIFAR-100, PACFL outperforms all SOTA methods by $+19\%$, $+18\%$, $+19\%$, $+18\%$ for FedAvg, FedProx, FedNova, SCAFFOLD) as well as all the personalized competitors (by $+27\%$, $+13\%$, $+1.5\%$, $+33\%$ for LG, PerFedAvg, IFCA, CFL ). We tuned the hyperparameters in each baseline to obtain the best results. IFCA achieved the best performance with 2 clusters which is consistent with the results in (Ghosh et al. 2020).
Within each cluster and sharing more training) which realizes the FedAvg baseline (pure globalization). On the contrary, group all clients into 1 cluster and the scenario reduces to PerFedAvg. This is precisely what PACFL is designed to do, to share in the training of the averaged global model to benefit from more personalization. When \( \beta \) is big enough, PACFL will take the advantage of a certain level of personalization (Li et al. 2021b). While these partitioning strategies synthetically simulate Non-IID data distributions in FL by partitioning a dataset into multiple smaller Non-IID subsets, they cannot design real and challenging Non-IID data distributions. According to the prior sections, due to the small intra-class distance (similarity between distribution of the classes) in the used benchmark datasets, all baselines benefited highly from globalization. This is the reason that PACFL and IFCA could achieve a high performance with only 2 clusters.

### How Many Clusters Are Needed?

As we emphasized in prior sections, one of the significant contributions of PACFL is that the server can easily determine the best fitting number of clusters just by analyzing the proximity matrix without running the whole federation. For instance, for IID scenarios, we expect the best fitting number of clusters to be one. The reason behind is that under IID setting, since all clients have similar distributions, they can share in the training of the averaged global model to benefit all. On the other hand, in the case of MIX-4, we expect the best fitting number of clusters to be four. More generally, we expect the best fitting number of clusters to be dependent on the similarities/dissimilarities in distributions among the clients.

### Globalization and Personalization Trade-off

To cope with the statistical heterogeneity, previous works incorporated a proximal term in local optimization or modified the model aggregation scheme at the server side to take the advantage of a certain level of personalization (Li et al. 2020; Vahidian, Morafah, and Lin 2021; Deng, Kamani, and Mahdavi 2020). Though effective, they lack the flexibility to trade off between personalization and globalization. Our proposed PACFL approach can naturally provide this globalization and personalization trade-off. Fig. 2 visualizes the accuracy performance behavior of PACFL versus different values of \( \beta \) which is the \( L_2 \) (Euclidean) distance between two clusters when the proximity matrix obtained as in Eq. 2, or Eq. 3. In other words, \( \beta \) is a threshold controlling the number of clusters as well as the similarity of the data distribution of clients within a cluster under Non-IID label skew. The blue curve and the red bars demonstrate the accuracy, and the number of clusters respectively for each \( \beta \). Varying \( \beta \) in a range which depends upon the dataset, PACFL can sweep from training a fully global model (with only 1 cluster) to training fully personalized models for each client.

As is evident from Fig. 2, the behaviour of PACFL on each dataset is similar. In particular, increasing \( \beta \) decreases the number of clusters (by grouping more number of clients within each cluster and sharing more training) which realizes more globalization. When \( \beta \) is big enough, PACFL will group all clients into 1 cluster and the scenario reduces to the FedAvg baseline (pure globalization). On the contrary, decreasing \( \beta \) increases the number of clusters, which leads to more personalization. When \( \beta \) is small enough, individual clusters would be formed for each client and the scenario degenerates to the SOLO baseline (pure personalization). As demonstrated, on all datasets, all clients benefit from some level of globalization. This is the reason why decreasing the number of clusters can improve the accuracy performance in comparison to SOLO. In general, finding the optimal trade-off between globalization and personalization depends on the level of heterogeneity of tasks, the intra-class distance of the dataset, as well as the data partitioning across the clients. This is precisely what PACFL is designed to do, to find this optimal trade-off before initiating the federation via the proximity matrix at the server. IFCA lacks this trade-off capability as it must define a fixed number of clusters \( (C > 1) \) or with \( C = 1 \) it would degenerate to FedAvg.

### Mixture of 4 Datasets

Existing studies have been evaluated on simple partitioning strategies, i.e., Non-IID label skew (20%) and (30%) (Li et al. 2021b). While these partitioning strategies synthetically simulate Non-IID data distributions in FL by partitioning a dataset into multiple smaller Non-IID subsets, they cannot design real and challenging Non-IID data distributions. According to the prior sections, due to the small intra-class distance (similarity between distribution of the classes) in the used benchmark datasets, all baselines benefited highly from globalization. This is the reason that PACFL and IFCA could achieve a high performance with only 2 clusters.

In order to better assess the potential of the SOTA baselines under a real-world and challenging Non-IID task where the local data of clients have strong statistical heterogeneity, we design the following experiment naming it as MIX-4. We assume that each client owns data samples from one of the four datasets, i.e., USPS (Hull 1994), CIFAR-10, SVHN, and FMNIST. In particular, we distribute CIFAR-10, SVHN, FMNIST, USPS among 31, 25, 27, 14 clients respectively (100 total clients) where each client receives 500 samples from all classes of only one of these dataset. This is a hard Non-IID task. We compare our approach with the SOTA baselines in the classification of these four different datasets, and we present the average of the clients’ final local test accuracy in Table 3. As can be seen, IFCA is unable to effectively handle this difficult scenario with tremendous data heterogeneity with just two clusters, as suggested in (Ghosh et al. 2020) as the best fitting number of clusters. IFCA (2) with 2 clusters performs almost as poorly as the global baselines while PACFL can find the optimal number of clusters in this task (four clusters) and outperforms all SOTA by a large margin. The results of IFCA (4) with 4 clusters is 76.79 \( \pm \) 0.43. As observed, PACFL surpasses all the global competitors (by +14%, +15%, +16%, +8% for FedAvg, FedProx, FedNova, SCAFFOLD) as well as all the personalized competitors (by +10%, +35%, +7%, +16% for LG, PerFedAvg, IFCA, CFL respectively). Further, the visualization in Fig. 3c and 3d also show how PACFL determines the optimal number of clusters on MIX-4.

### Table 2: Test accuracy comparison across different datasets for Non-IID label skew (20%).

| Algorithm | FMNIST | CIFAR-10 | CIFAR-100 | SVHN |
|-----------|--------|----------|-----------|------|
| SOLO      | 95.92 ± 0.57 | 79.22 ± 1.67 | 32.28 ± 0.23 | 79.72 ± 1.37 |
| FedAvg    | 77.3 ± 4.9 | 49.8 ± 3.3 | 53.73 ± 0.50 | 80.2 ± 0.8 |
| FedProx   | 74.9 ± 2.6 | 50.7 ± 1.7 | 54.35 ± 0.84 | 79.3 ± 0.9 |
| FedNova   | 70.4 ± 5.1 | 46.5 ± 3.5 | 53.61 ± 0.42 | 75.4 ± 4.8 |
| SCAFFOLD  | 42.8 ± 28.7 | 49.1 ± 1.7 | 54.15 ± 0.42 | 62.7 ± 11.6 |
| LG        | 96.80 ± 0.51 | 86.31 ± 0.82 | 45.98 ± 0.34 | 92.61 ± 0.45 |
| PerFedAvg | 95.35 ± 1.15 | 85.46 ± 0.56 | 60.19 ± 0.15 | 93.32 ± 2.05 |
| IFCA      | 97.15 ± 0.01 | 87.99 ± 0.15 | 71.84 ± 0.23 | 95.42 ± 0.06 |
| CFL       | 77.93 ± 2.19 | 51.11 ± 0.10 | 40.29 ± 2.73 | 73.62 ± 1.76 |
| PACFL     | 97.54 ± 0.08 | 89.30 ± 0.41 | 73.10 ± 0.21 | 95.77 ± 0.18 |

10048
Table 3: The benefits of PACFL are particularly pronounced when the tasks are extremely Non-IID. This table evaluates 4 in terms of the top-1 test accuracy performance. While the results clearly show that PACFL is very robust even under such difficult data heterogeneity scenarios.

| Algorithm   | MIX-4     | Algorithm   | MIX-4     |
|-------------|-----------|-------------|-----------|
| SOLO        | 55.08 ± 0.29 | LG          | 58.49 ± 0.46 |
| FedAvg      | 63.68 ± 1.64 | PerFedAvg   | 42.60 ± 0.60 |
| FedProx     | 61.86 ± 3.73 | IFCA (2)    | 70.32 ± 3.57 |
| FedNova     | 60.92 ± 3.60 | CFL         | 61.18 ± 2.63 |
| Scaffold    | 69.26 ± 0.84 | PACFL       | 77.83 ± 0.33 |

Generalization to Newcomers

PACFL provides an elegant approach to handle newcomers arriving after the federation procedure, to learn their personalized model. In general, for all other baselines it is not clear how they can be extended to handle clients unseen at training (federation). We show in Algorithm 3 how PACFL can simply be generalized to handle clients arriving after the end of federation, to learn their personalized model. The unseen client will send the signature of its data to the server and the server determines which cluster it belongs to. The server then sends the corresponding model to the newcomer and the newcomer fine tunes the received model. To evaluate the quality of newcomers’ personalized models, we design an experiment under Non-IID label skew (20%) where only 80 out of 100 clients are participating in a federation with 50 rounds. Then, the remaining 20 clients join the network at the end of federation and receive the model from the server and personalize it for only 5 epochs. The average final local test accuracy of the unseen clients is reported in Table 4. Focusing on CIFAR-100, as observed, some of the personalized baselines including LG and PerFedAvg perform as poor as global baselines and SOLO. PACFL consistently outperforms other baselines by a large margin.
where a limited amount of communication round is permissible for federation under a heterogeneous setup. To this end, we compare the performance of the proposed federated learning (FL) algorithms with baselines and present the average of final local test accuracy over all clients versus number of communication rounds. Baselines and the communication rounds budget of 80 is considered in this work. Figure 4(a) to 4(c) show the test accuracy versus number of communication rounds for Non-IID (20%) across CIFAR-10, CIFAR-100, and SVHN datasets. Our proposed federated learning (FL) algorithms consistently outperform strong competitors. PACFL converges fast to the desired accuracy and consistently outperforms strong competitors.

Conclusion

In this paper, we proposed a new framework for clustered FL, that directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces spanned by their principal vectors. This approach provides a simple, but yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews.
References

Bonawitz, K. A.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, 1175–1191. ACM.

Day, W. H.; and Edelsbrunner, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. Journal of classification, 1(1): 7–24.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive Personalized Federated Learning. CoRR, abs/2003.13461.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. Advances in Neural Information Processing Systems, 33: 3557–3568.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An Efficient Framework for Clustered Federated Learning. In Advances in Neural Information Processing Systems 33.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. The Journal of Machine Learning Research, 13(1): 723–773.

Haddadpour, F.; and Mahdavi, M. 2019. On the convergence of local descent methods in federated learning. arXiv preprint arXiv:1910.14425.

Hallac, D.; Leskovec, J.; and Boyd, S. P. 2015. Network Lasso: Clustering and Optimization in Large Graphs. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, 387–396. ACM.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 770–778. IEEE Computer Society.

Hershey, J. R.; and Olsen, P. A. 2007. Approximating the Kullback-Leibler Divergence Between Gaussian Mixture Models. In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07, volume 4, IV–317–IV–320.

Hull, J. J. 1994. A database for handwritten text recognition research. IEEE Transactions on pattern analysis and machine intelligence, 16(5): 550–554.

Jain, P.; Netrapalli, P.; and Sanghavi, S. 2013. Low-rank matrix completion using alternating minimization. In Boneh, D.; Roughgarden, T.; and Feigenbaum, J., eds., Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1–4, 2013, 665–674. ACM.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In Proceedings of the 37th International Conference on Machine Learning, ICML, volume 119, 5132–5143. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Toronto.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4): 541–551.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021a. Federated Learning on Non-IID Data Silos: An Experimental Study. CoRR, abs/2102.02079.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021b. Federated learning on non-iid data silos: An experimental study. arXiv preprint arXiv:2102.02079.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, March 2-4, 2020. mlsys.org.

Liang, P. P.; Liu, T.; Ziyin, L.; Salakhutdinov, R.; and Morency, L.-P. 2020a. Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523.

Liang, P. P.; Liu, T.; Ziyin, L.; Salakhutdinov, R.; and Morency, L.-P. 2020b. Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523.

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three Approaches for Personalization with Applications to Federated Learning. CoRR, abs/2002.10619.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, 1273–1282. PMLR.

McMahan, B.; and Ramage, D. 2017. Federated learning: Collaborative machine learning without centralized training data. Google Research Blog, 3.

Mendieta, M.; Yang, T.; Wang, P.; Lee, M.; Ding, Z.; and Chen, C. 2022. Local learning matters: Rethinking data heterogeneity in federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8397–8406.

Morafah, M.; Vahidian, S.; Wang, W.; and Lin, B. 2022. FLIS: Clustered Federated Learning via Inference Similarity for Non-IID Data Distribution. In NeurIPS Workshop.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning.

Qian, G.; Sural, S.; Gu, Y.; and Pramanik, S. 2004. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In Proceedings of the 2004 ACM symposium on Applied computing, 1232–1237.

Rad, H. J.; Abdizadeh, M.; and Szabó, A. 2021. Federated Learning with Taskonomy for Non-IID Data. CoRR, abs/2103.15947.

Sarcheshmehpour, Y.; Leinonen, M.; and Jung, A. 2021. Federated Learning from Big Data Over Networks. In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6–11, 2021, 3055–3059. IEEE.

Sattler, F.; Müller, K.; and Samek, W. 2021. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. IEEE Trans. Neural Networks Learn. Syst., 32(8): 3710–3722.

Talwalkar, A.; Kumar, S.; Mohri, M.; and Rowley, H. 2013. Large-scale SVD and manifold learning. The Journal of Machine Learning Research, 14(1): 3129–3152.

Vahidian, S.; Morafah, M.; Chen, C.; Shah, M.; and Lin, B. 2022. Rethinking Data Heterogeneity in Federated Learning: Introducing a New Notion and Standard Benchmarks. In NeurIPS 2022 workshop.

Vahidian, S.; Morafah, M.; and Lin, B. 2021. Personalized federated learning by structured and unstructured pruning under data heterogeneity. IEEE ICDCS.
Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In Advances in Neural Information Processing Systems, volume 33, 7611–7623. Curran Associates, Inc.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. arXiv preprint arXiv:1806.00582.