Asynchronous Personalized Federated Learning with Irregular Clients

Zichen Ma ZICHENMA1@LINK.CUHK.EDU.CN
Yu Lu YULU1@LINK.CUHK.EDU.CN
JD AI Research, Beijing, China
The Chinese University of Hong Kong, Shenzhen, Guangdong, China
Wenye Li WYLI@CUHK.EDU.CN
Shuguang Cui SHUGUANGCUI@CUHK.EDU.CN
The Chinese University of Hong Kong, Shenzhen, Guangdong, China

Editors: Emtiyaz Khan and Mehmet Gönen

Abstract

To provide intelligent and personalized models for clients, personalized federated learning (PFL) enables learning from data, identifying patterns, and making automated decisions in a privacy-preserving manner. PFL involves independent training for multiple clients with synchronous aggregation steps. However, the assumptions made by existing works are not realistic given the heterogeneity of clients. In particular, the volume and distribution of collected data vary in the training process, and the clients also vary in their available system configurations, which leads to vast heterogeneity in the system. To address these challenges, we present an *asynchronous* method (AsyPFL), where clients learn personalized models w.r.t. local data by making the most informative parameters less volatile. The central server aggregates model parameters asynchronously. In addition, we also reformulate PFL by unifying both synchronous and asynchronous updating schemes with an asynchrony-related parameter. Theoretically, we show that AsyPFL's convergence rate is state-of-the-art and provide guarantees of choosing key hyperparameters optimally. With these theoretical guarantees, we validate AsyPFL on different tasks with non-IID and staleness settings. The results indicate that, given a large proportion of irregular clients, AsyPFL excels at empirical performance compared with vanilla PFL algorithms on non-IID and IID cases.

Keywords: Personalized federated learning; Asynchronous optimization

1. Introduction

The recently emerged personalized federated learning involves collaboratively training personalized models on clients without disclosing their private datasets (Smith et al. (2017)). It aims to produce highly accurate statistical models by aggregating knowledge from disparate data sources. Specifically, the processes of this paradigm can be summarized as follows: (i) the server sends the global model (or the initialization of the global model) to all clients in a round; (ii) the clients train personalized models w.r.t. their local data samples. (iii) the server collects newly trained models to update the global model synchronously and broadcasts the new model to clients. The process is repeated until convergence (Mansour et al. (2020)).

However, to deploy PFL in practice, the resulting system must be accurate and satisfy several constraints, such as systemic and statistical heterogeneity. Unfortunately, simultaneously satisfying these varied constraints can be exceptionally difficult (Kairouz et al. (2019)).

Systemic heterogeneity exists in clients with limited computing and communication resources, where clients may fail to communicate with the server, i.e., be inactive or return updates asynchronously. These undesirable behaviors are proven to degrade the system's performance since they magnify the discrepancies between irregular and global models (Sahu et al. (2018); Chen et al. (2018)). Statistical heterogeneity refers to non-IID data, where data are highly skewed and imbalanced and vary across clients (McMahan et al. (2017)). From a statistical perspective, it leads to distribution shifts, which raises the difficulties of model convergence.

Many prior efforts have considered a partial client participation scheme to relieve the heterogeneity concern, where the server only aggregates local models without waiting for irregular clients. Unfortunately, only a few client selection policies have been proven to work in this setting, and the selection must be independent of the status of clients (Li et al. (2019a)). In other words, for the training to converge successfully, all selected clients must be able to train their personalized models and upload the results whenever they are selected. Thus, the traditional PFL paradigm requires participating clients to be dedicated to the training during the entire period.

PFL typically takes thousands of communication rounds to converge. Ensuring that all clients will be available during the entire training in practice is not easy. Moreover, multiple apps typically run simultaneously on clients, competing for already highly constrained hardware resources. As such, as expected, the system cannot guarantee that clients will complete their assigned training tasks in every training round. Even for cross-silo applications, where the system may adopt more powerful computers or cloud servers, clients' availability can still be an issue due to the increasingly widespread usage of preemptive cloud services such as AWS's spot instances, where the user process can be interrupted unexpectedly.

While many other methods have been proposed to mitigate the workload of individual clients, such as weight compression and federated dropout (Caldas et al. (2018)), they cannot remove the possibility of confronting irregular client behaviors during the training. This probability, intuitively, increases as more clients join the training. Therefore, in large-scale PFL, many low-end clients have to be excluded from joining the system in the first place, which restricts the potential availability of training datasets and weakens the system's applicability. Furthermore, the existing training procedure does not specify how to react when confronting undesirable client behaviors such as asynchronous communication and does not analyze such behaviors' adverse effects on the training progress.

Inspired by the critical roles of asynchronous optimization in several business applications of AI services (Bianchi et al. (2015)), we address these constraints by presenting theoretical analysis and proposing a new PFL scheme, where model aggregations are asynchronous, and the most informative parameters of models are less volatile during training. This scheme allows irregular clients to be involved in the system and optimized to enhance performance.

Contributions of this paper are summarized as follows. First, we formulate a new asynchronous method designed for PFL (AsyPFL) by incorporating an elastic term into the local objective to stabilize the training of personalized models. The asynchronous structure of AsyPFL has a crucial advantage: it allows asynchronous updates from multiple clients and is robust against high communication delays with an asynchrony-related parameter. Thus, AsyPFL mitigates the irregular client behavior caused by two-fold heterogeneity.

Second, we exploit the convexity-preserving and smoothness-enabled properties of AsyPFL to facilitate the convergence analysis of the method. We present new assumptions for non-IID data to measure and bound their impact using model discrepancies. We also analyze the bi-level training strategy for choosing proper key hyperparameters. With optimized hyperparameters, AsyPFL can obtain the state-of-the-art speedup (resp. sublinear speedup of order 2/3), compared with the current works with linear speedup (resp. sublinear speedup of order 1/2), for the strongly convex objective.

Finally, we supplement our theoretical findings with extensive corroborating experimental results that demonstrate the superiority of the proposed scheme over commonly used PFL algorithms. We empirically evaluate the performance of AsyPFL using real and synthetic datasets that capture the statistical diversity of clients' data and show that AsyPFL outperforms the vanilla PFL methods in dealing with irregular clients.

2. Related Work

Personalized Federated Learning Given the variability of data in federated learning, personalization is a natural approach used to improve accuracy, and numerous works have been proposed for PFL. Mainly, Smith et al. (2017) first explores PFL via a primal-dual multi-task learning framework, which applies to convex settings. As summarized in (Deng et al. (2020); Tan et al. (2021)), the subsequent works have explored PFL through local customization (Fallah et al. (2020b); Jiang et al. (2019); Khodak et al. (2019); Mansour et al. (2020); Wang et al. (2019)), where personalized models are built by customizing a well-trained global model. There are several ways to conduct customization: (i) mixture of the global and local models customizes for each client by combining the global model with the client's latent local model (Hanzely and Richtárik (2020); Deng et al. (2020); Mansour et al. (2020)); (ii) meta-learning approaches build an initial meta-model that can be updated effectively by Hessian or its approximations. The personalized models are customized w.r.t. local data (Fallah et al. (2020b); Nichol et al. (2018); Fallah et al. (2020a); Khodak et al. (2019); Jiang et al. (2019)); (iii) local fine-tuning methods customize the global model using local datasets to learn personalized models for each client (Mansour et al. (2020); Liang et al. (2020)).

Asynchronous Optimization Asynchronous optimization can accelerate training performance in near-linear time but is known to have a staleness effect (Mitliagkas et al. (2016); Hadjis et al. (2016); Lian et al. (2015); Cui et al. (2016)), which is introduced by delayed gradients. Moreover, the method suffers from high computational complexity in model dynamics, i.e., hyperparameters tuning and the trade-off between speed and accuracy (Hakimi et al. (2019); Zhang et al. (2015)). To address these problems, dynamically changing learning rates, including learning rate decay and adaptive learning rates, are proposed (Zheng et al. (2017); Dai et al. (2018)). Furthermore, Hakimi et al. (2019) models the delay with a

new metric and optimizes performance by minimizing the gap to tackle the staleness caused by momentum-based approaches. However, these works only focus on the distributed system but little on federated learning, where clients are heterogeneous.

3. Asynchronous Personalized Federated Learning (AsyPFL)

3.1. AsyPFL: Preliminaries and Problem Formulation

Notations Given M clients and a server in the system and the k-th client has m_k data samples, where $m = \sum_{k=1}^{M} m_k$ be the total number of samples among M clients. Denote by $X = \cup_k X_k$ the set of data among M clients, $\omega \in \mathbb{R}^{d_0}$ the shared parameters, $\beta = (\beta_1, \beta_2, ..., \beta_M)$ with $\beta_k \in \mathbb{R}^{d_k}$ the personalized local models in the k-th client, $F_k : (\mathbb{R}^{d_0|d_k}, X_k) \to \mathbb{R}$ the expected loss over the data distribution of the k-th client, ΔT_k^g the time cost of one update in local optimizer, and ΔT_k^c the time cost for one communication between the k-th client and the server, respectively.

In PFL, clients communicate with the server to solve

$$\min_{\beta} F(\beta) = \sum_{k=1}^{M} \frac{m_k}{m} F_k(\omega, \beta_k) \tag{1}$$

to find personalized models β. Generally, Equation (1) is optimized w.r.t. β using stochastic gradient descent (SGD) in clients. The server updates the model synchronously, i.e., it has to wait until all participants send model updates to the server. However, this scheme is inefficient, given heterogeneous settings.

Instead of synchronously solving the traditional PFL problem in Equation (1), AsyPFL takes a different approach by employing a regularized local objective and an asynchronous updating scheme, where

$$\begin{aligned} \min_{\beta_k} \widetilde{F}_k(\beta_k; \omega^*) &= F_k(\beta_k) + \frac{\lambda}{2}(\beta_k - \omega^*)^T \mathrm{diag}(h(\omega^*))(\beta_k - \omega^*) \\ s.t. \;\; \omega^* &\in \arg\min_{\omega} \sum_{k=1}^{M} p^k F_k(\omega). \end{aligned} \tag{2}$$

λ is the regularization weight that controls the interpolation between the global and personalized models. When $\lambda = 0$, the local objective aims at training purely local models. $h(\cdot)$ denotes the Fisher information matrix, which is the negative expected Hessian of the log-likelihood function, and $\mathrm{diag}(h(\cdot))$ is the matrix that preserves the Fisher information matrix's diagonal values, penalizing parts of the parameters that are too volatile in a round. $p^k = \frac{m_k}{m}$.

In AsyPFL, while ω^* is found by exploiting the data aggregation from multiple clients at the outer level, β_k is optimized w.r.t. k-th client's data and is maintained at a bounded distance from ω^* at the inner level. This bi-level objective decouples the process of optimizing personalized models from learning the global model, which preserves the low complexity.

AsyPFL incorporates an asynchronous updating scheme that allows each client to update its model or download the global model whenever it is ready. Denote by $\{\omega_k^\tau\}_{\tau=0}^{T}$ the consecutive models trained in k-th client from step 0 to T, and π_k the staleness indicator of the

k-th client, which represents the communication schedule. Let $\Pi_k := \{l\pi_k E | l = 0, 1, 2, 3, ...\}$ denote the set of time steps when the k-th client communicates with the server, i.e., the k-th client communicates with the server every $\pi_k E$ step. As a result, when $\tau \in \Pi_k$, the k-th client uploads its updates and downloads the newly aggregated model from the server. Otherwise, it continues to train the model locally. we can formulate this asynchronous update rule as

$$\omega_k^{\tau+1} = \begin{cases} \omega_k^\tau - \eta_\tau \nabla F_k(\omega_k^\tau), & \text{if } \tau + 1 \notin \Pi_k \\ \omega^{\tau+1}, & \text{if } \tau + 1 \in \Pi_k. \end{cases} \tag{3}$$

Equation (3) unifies the two updating schemes using Π_k, i.e., when $\Pi_1 = \Pi_2 =, ..., = \Pi_M$, the clients communicate with the server simultaneously. Otherwise, it is the asynchronous case.

Finally, the server aggregates the global model by

$$\omega^{\tau+1} = \omega^\tau + \sum_{k=1,...,M; \tau+1 \in \Pi_k} p^k (\omega_k^{\tau+1} - \omega_k^{r_k^\tau}), \tag{4}$$

where $\omega_k^{\tau+1}$ is the uploaded model trained from the initialization of $\omega_k^{r_k^\tau}$ in the k-th client, and $\omega_k^{r_k^\tau} = \omega^{r_k^\tau}$ is the global model downloaded by the client in the previous round. Denote by $r_k^\tau \in \{0, 1, ..., T\}$ the corresponding time step when the k-th client downloads the global model $\omega^{r_k^\tau}$.

Assumption 1 *(L_c-continuous and L_s-smooth functions) F_k is L_c-continuous and L_s-smooth for $k = 1, 2, ..., M$.*

Assumption 2 *(μ-strongly convexity) F_k is μ-strongly convex with constant $\mu > 0$ for $k = 1, 2, ..., M$. $\forall a, a^{'}$, we have:*

$$F_k(a^{'}) - F_k(a) - \nabla F_k(a)||a^{'} - a|| \geq \frac{\mu}{2}||a^{'} - a||^2$$

Assumption 3 *(Bounded variance) Suppose ξ_k^τ is a uniformly sampled data point from X_k at $\tau = 1, 2, ..., T$, and $k = 1, 2, ..., M$. The variance of stochastic gradients in each client is bounded by*

$$\mathbb{E}[||\nabla F_k(\omega_k^\tau, \xi_k^\tau) - \nabla F_k(\omega_k^\tau)||^2] \leq \sigma_k^2,$$

Assumption 4 *(Bounded gradient discrepancy) Let $\omega_k^\tau = \omega^\tau$ for $\tau = 1, 2, ..., T$, and $k = 1, 2, ..., T$. The discrepancy of model gradients is bounded by*

$$\max \mathbb{E}[||g_k^\tau - \overline{g}^\tau||] = \chi,$$

where $g_k^\tau = \nabla F_k(\omega_k^\tau, \xi_k^\tau)$ is the local stochastic gradient of sampled data ξ_k^τ, and $\overline{g}^\tau = \sum_{k=1}^M p^k \nabla F_k(\omega_k^\tau)$ is the expectation of gradients.

Assumption 5 *(Bounded model discrepancy) Denote by $\omega_k^* = \arg\min F_k(\omega)$ the optimal model in k-th client, and ω^0 the initialization of the global model. For a given ratio $q \gg 1$, the discrepancy between ω^0 and ω^* is sufficiently larger than the discrepancy between ω_k^* and ω^*, i.e., $||\omega^0 - \omega^*|| > q||\omega_k^* - \omega^*||$*

Algorithm 1 AsyPFL. M clients are indexed by k, λ is the regularization weight, η and δ are the learning rate, and γ is the decay factor; T is the maximal number of communication rounds; Q represents the waiting queue; ΔT_k^g denotes the time cost of one update in the local optimizer; ΔT_k^c is the time cost for one communication between the k-th client and the server.

Server executes:

initialize local epoch E, ω^0, β_k^0.
for $\tau = 0, 1, ..., T$ **do**
  $S^\tau \leftarrow$ (subset of M clients).
  for each client $k \in S^\tau$ **in parallel do**
    $\omega_k^{\tau+1}$,r_k^τ,ΔT_k^g,$\Delta T_k^c \leftarrow$ ClientUpdate(k,ω^τ,E_k).
    Push ($\omega_k^{\tau+1}$,r_k^τ,ΔT_k^g,ΔT_k^c) into Q.
    $\overline{\pi}_{min} = \min\{\frac{\Delta T_k^c}{\Delta T_k^g p^k}\}$, $\pi_k = |\frac{\Delta T_k^c}{\Delta T_k^g p^k \overline{\pi}_{min}}|$.
    $E_k = \pi_k E$
    $\omega^{\tau+1} = \omega^\tau + \sum_{k;\tau+1\in\Pi_k} p^k(\omega_k^{\tau+1} - \omega_k^{r_k^\tau})$
  end for
end for

ClientUpdate(k,ω^τ,E_k):

$r_k^\tau = \tau$, $E = E_k$.
for $e \in \{1, 2, ..., E\}$ **do**
  $\omega_k^{\tau+1} = \omega_k^\tau - \frac{\eta}{1+\gamma\tau}\nabla F_k(\omega_k^\tau)$.
  $\beta_k^{\tau+1} = \beta_k^\tau - \frac{\delta}{1+\gamma\tau}(\nabla F_k(\beta_k^\tau) + \lambda(\beta_k^\tau - \omega_k^{\tau+1})diag(h(\omega_k^{\tau+1})))$.
  Calculate the average ΔT_k^g, ΔT_k^c.
end for

While Assumptions 1 and 2 are standard for convergence analysis, Assumptions 3 and 4 are widely used in the FL context in which σ^2 and χ quantify the sampling noise and the diversity of the client's data distribution, respectively (Karimireddy et al. (2020); Fallah et al. (2020b); Li et al. (2019b); Yu et al. (2019)). Note that we avoid using the uniformly bounded gradient assumption, i.e., $||\nabla F_k(a) \leq G||$, $\forall k$, used in several related works (Deng et al. (2020); Fallah et al. (2020b)). It is shown that this assumption is not satisfied in the strongly convex minimization (Zhou et al. (2019); Li et al. (2019b)).

3.2. AsyPFL: Algorithm

In this section, we propose AsyPFL, presented in Algorithm 1, to solve Equation (2). Specifically, the algorithm jointly optimizes the global model ω and personalized models β in an alternating fashion. Optimization proceeds in two phases: (i) updates to the global model, ω, are computed across the network, and then (ii) the personalized models β_k are fit on each local client. Optimizing ω is different from conventional PFL algorithms, where local epochs E_k are carefully chosen by calculating the staleness indicator π_k on the k-th client.

In AsyPFL, (i) the server first broadcasts the newly aggregated global model (or initialization of the global model) and the number of local epochs to participants in the τ-th

round; (ii) the k-th client runs a local optimizer w.r.t. local datasets on ω_k and β_k with a decayed learning rate. The algorithm solves the local subproblem of $\min_\omega \sum_{k=1}^{M} p^k F_k(\omega)$ approximately. For personalization, client k solves the global-regularized local objective $\min_{\beta_k} \widetilde{F}_k(\beta_k; \omega^*)$ inexactly at each round. We note that another natural choice to solve Equation (2) is first to obtain ω^*, and then for each client k, perform finetuning on the local objective $\min_{\beta_k} \widetilde{F}_k(\beta_k; \omega^*)$. These two approaches will arrive at the same solutions in strongly convex cases. In non-convex settings, we observe that an approximate solver in joint optimization has additional benefits. Empirically, we find that the updating scheme tends to guide the optimization trajectory towards a better solution compared with finetuning starting from ω^*; (iii) the server collects the time cost of updates both in the local optimizer and communications to adjust the staleness indicator and local epochs dynamically. Since clients communicate with the server asynchronously, the server aggregates and updates the global model according to Equation (4).

3.3. AsyPFL: Theoretical Analysis

In this section, we theoretically analyze the asynchronous scheme used in AsyPFL, which decouples the optimization of personalized model β from the global model ω. In the following analysis, we first present the convergence bounds with/without decayed learning rates. Then we analyze the method to choose optimal hyperparameters such as local epochs E_k and staleness indicator π_k on the k-th client.

Theorem 1 *Given the maximum time step T, for any fixed number of local epoch E, the fixed learning rate $\eta \leq \frac{1}{4L}$, F^* and F_k^* denotes the minimum of objective functions F and F_k. Under Assumption 1-4, the error of AsyPFL is bounded by*

$$\mathbb{E}[F(\omega^T)] - F^* \leq \frac{L(1-\mu\eta)^T}{2}(||\omega^0 - \omega^*||^2 - 2B) + B, \tag{5}$$

where

$$B = \frac{L\eta}{2\mu}(\sum_{k=1}^{M} p^{2k}\sigma_k^2\Sigma_k^2 + 6L\Gamma + 2E\sum_{k=1}^{M} p^k(\pi_k^2 E - \pi_k)\chi^2),$$

and $\Gamma = |F^* - \sum_{k=1}^{M} p^k F_k^*|$.

Theorem 1 presents the error of AsyPFL with a fixed learning rate. Inspired by (Li et al. (2019a)), convergence can be guaranteed with learning rate decay for synchronous PFL in non-IID settings, which may also be applicable for asynchronous models. We theoretically analyze and compare the convergence with learning rate decay in Theorem 2.

Theorem 2 *Given the same conditions in Theorem 1 except for the fixed learning rate, $\gamma = \max\{8\frac{L}{\mu}, \max_k\{\pi_k\}E\}$, and decayed learning rate $\eta_\tau = \frac{2}{\mu(\gamma+\tau)}$, the error of AsyPFL is bounded by*

$$\mathbb{E}[F(\omega^T)] - F^* \leq \frac{2L}{\mu(\gamma+T)^{2/3}}(\frac{A}{\mu} + 2L||\omega^0 - \omega^*||), \tag{6}$$

where

$$A = \sum_{k=1}^{M} p^{2k}\sigma_k^2\Sigma_k^2 + 6L\Gamma + 8E\sum_{k=1}^{M} p^k(\pi_k^2 E - \pi_k)\chi^2.$$

Table 1: Convergence results of several PFL algorithms are summarized. The maximum number of communications is T. SC refers to strongly convex, and NC refers to non-convex.

Algorithm	Bounded gradient	Convexity	Convergence speedup
Karimireddy et al. (2020)	✓	μ-SC	$O(\frac{1}{\sqrt{T}})$
Deng et al. (2020)	×	NC	$O(\frac{1}{T^{1/3}})$
Dinh et al. (2020)	✓	μ-SC	$O(\frac{1}{\sqrt{T}})$
Li et al. (2020)	×	NC	$O(\frac{1}{\sqrt{T}})$
AsyPFL	×	μ-SC	$O(\frac{1}{T^{2/3}})$

Theorems 1 and 2 demonstrate the convergence bounds of AsyPFL with fixed and decayed learning rates, respectively. Theorem 2 confirms that under the non-IID and asynchronous cases, AsyPFL reaches a state-of-the-art convergence rate with a sublinear speed-up of $O(1/T^{2/3})$ using decayed learning rate. We compare our results with several PFL algorithms in Table 1.

Next, we present how to achieve the optimal local epochs E in Theorem 3 and optimal staleness indicator π in Theorem 4.

Theorem 3 *Denote by $T_{\min}$ the minimal number of global epochs to achieve ϵ-accuracy, where*

$$\mathbb{E}[\omega^{T_{\min}})] - F^* \leq \frac{2L}{\mu(\gamma + T_{\min})^{2/3}}(\frac{A}{\mu} + 2L||\omega^0 - \omega^*||) \leq \epsilon.$$

Given the same conditions as Theorem 2, the minimum communication round $R_{\min}(E) = \frac{T_{\min}}{E}$ can be achieved by

$$E = \sqrt{\frac{V_\epsilon + \frac{4L^2}{\mu\epsilon}||\omega^0 - \omega^*||}{\frac{16L}{\mu^2\epsilon}\chi^2 \sum_{k=1}^M p^k \pi_k^2}},$$

where $V_\epsilon = \frac{2L}{\mu\epsilon}\frac{\sum_{k=1}^M p_k^2\sigma_k^2 + 6L\Gamma}{\mu}$.

Theorem 3 gives the optimal choice of E to reduce communication overhead with the non-IID data, i.e., $\chi \neq 0$. The choice of E is not static throughout the training process. It can be seen that E is proportional to the global error $||\omega^0 - \omega^*||$, which indicates that E can be reduced as the training continues.

To further estimate the overall training time for achieving ϵ-accuracy, we define the total time cost of k-th client as

$$C_{\pi_k} = \Delta T_k^g T_{\min} + \Delta T_k^c \frac{R_{\min}}{\pi_k} = \Delta T_k^g T_{\min} + \Delta T_k^c \frac{T_{\min}}{E\pi_k}, \tag{7}$$

and the total time cost of the system is $\Delta T = \max\{C_{\pi_1}, C_{\pi_2}, ..., C_{\pi_k}\}$.

Theorem 4 *Given Assumptions 1-4, the minimal time cost $\min \Delta T$ is achieved by*

$$\pi_k = (\frac{Z_k \mu\epsilon\mu^2}{32LE^3\chi^2 p^k} \frac{\Delta T_k^c}{\Delta T_k^g})^{1/3},$$

Table 2: Statistics of datasets with non-IID partitions are summarized. The number of clients, the number of samples, and the mean and the standard deviation of data on each client are summarized.

Dataset	# clients	# Samples	# classes	Mean	Std.
MNIST	100	58,254	10	583	146
EMNIST	500	131,600	62	263	93
CIFAR100	100	59,137	100	591	32
Shakespeare	132	359,016	132	2,719	204
Sentiment140	1,503	90,110	2	60	41

where

$$Z_k = V_\epsilon + \frac{4L^2}{\mu\epsilon}||\omega^0 - \omega^*|| - \gamma + E\frac{16L}{\epsilon\mu^2}\sum_{l=1}^{M} p^l(\pi_l^2 E - \pi_l)\chi^2 - E\frac{16L}{\epsilon\mu^2}p^k(\pi_k^2 E - \pi_k)\chi^2.$$

Theorem 4 presents how to choose the optimal π to reduce time cost. Specifically, when the k-th client has a slow connection, i.e., ΔT_K^c is large, we can reduce communication rounds by setting a more significant π_k. When it has a limited computation resource, i.e., ΔT_k^g is large, we choose a smaller π_k. Furthermore, when the k-th client has a large data size, we choose a smaller π_k to avoid degrading the training quality. Please refer to supplemental material for the detailed proof of Theorems 1-4.

4. Experimental Results and Discussion

In this section, we first demonstrate the effectiveness and efficiency of AsyPFL in non-IID and IID data settings and compare it with several baseline algorithms. Then, we show the robustness of AsyPFL on irregular clients' challenges.

4.1. Experimental Setup

Datasets AsyPFL is evaluated on several benchmark datasets and compared with leading baselines. We use five benchmark datasets, which can be categorized as follows:

- *Image Classification* we adopt MNIST (LeCun et al. (1998)), EMNIST (Cohen et al. (2017)) dataset with Resnet50 (He et al. (2016)), CIFAR100 dataset (Krizhevsky et al. (2009)) with VGG11 (Simonyan and Zisserman (2014)) network. The MNIST dataset has images of hand-written digits from 100 clients with $58k$ samples. The EMNIST dataset contains images of hand-written digits and characters from 500 clients with a total of $131k$ samples, and CIFAR100 dataset contains $59k$ samples separated into 100 clients. For the IID setting, we split the training data randomly into equally sized shards and assigned one shard to every client. For the non-IID (m) setting, we assign every client sample from exactly m classes of the dataset. The data splits are non-overlapping and balanced, so every client has the same number of data points.

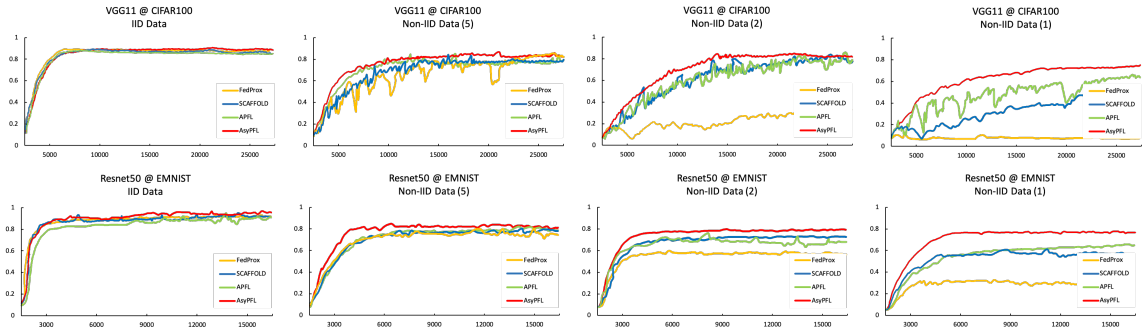

Figure 1: Testing Accuracy-Communication Rounds comparisons of VGG11 on CIFAR100 and Resnet50 on EMNIST in a distributed setting for IID and non-IID data. In the non-IID cases, every client only holds examples from exactly m classes in the dataset. All methods suffer from degraded convergence speed in the non-IID situation, but AsyPFL is affected the least.

- *Natural Language Processing (NLP)* We evaluate AsyPFL on Shakespeare dataset with an LSTM (McMahan et al. (2017)) to predict the next character, and Sentiment140 dataset (Go et al. (2009)) with an LSTM to classify sentiment. The Shakespeare dataset contains $359k$ samples separated into 132 clients, and the Sentiment140 dataset has $90k$ samples among 1,503 clients.

Table 2 summarizes the statistics of the datasets.

Metrics We evaluate AsyPFL and report the testing accuracy and best-mean-testing accuracy (BMTA) in IID and non-IID settings. The mean testing accuracy is the average testing accuracy of all clients.

Baselines We compare AsyPFL with several leading baselines in PFL. FedProx (Li et al. (2018)) incorporates a proximal term in local objective to improve the model performance on the non-IID data, SCAFFOLD adopts control variate to alleviate the effects of data heterogeneity (Karimireddy et al. (2020)), and APFL learns personalized local models to mitigate heterogeneous data on clients (Deng et al. (2020)). FedGATE introduces a local gradient tracking scheme to mitigate the heterogeneity (Haddadpour et al. (2021)), VRL-SGD eliminates the dependency on the gradient variance among clients (Liang et al. (2019)), and FedAMP employs federated attentive message passing to facilitate similar clients to collaborate more (Huang et al. (2021)).

4.2. Performance Comparison

Figure 1 shows the convergence comparison of gradient evaluations for the two models using different algorithms.

We observe that while all methods achieve comparably fast convergence in gradient evaluations on IID data, they suffer considerably in the non-IID setting. From left to right, as data becomes more non-IID, convergence worsens for FedProx, and it can sometimes diverge. SCAFFOLD and APFL exhibit their ability to alleviate the data heterogeneity,

Table 3: BMTA for the non-IID data setting

Methods	MNIST	CIFAR100	Sentiment140	Shakespeare
FedAvg	98.30	2.27	59.14	51.35
FedGATE	**99.15**	80.94	68.84	54.71
VRL-SGD	98.86	2.81	68.62	52.33
APFL	98.49	77.19	68.81	55.27
FedAMP	99.06	81.17	**69.01**	58.42
AsyPFL	99.10	**81.38**	68.95	**60.49**

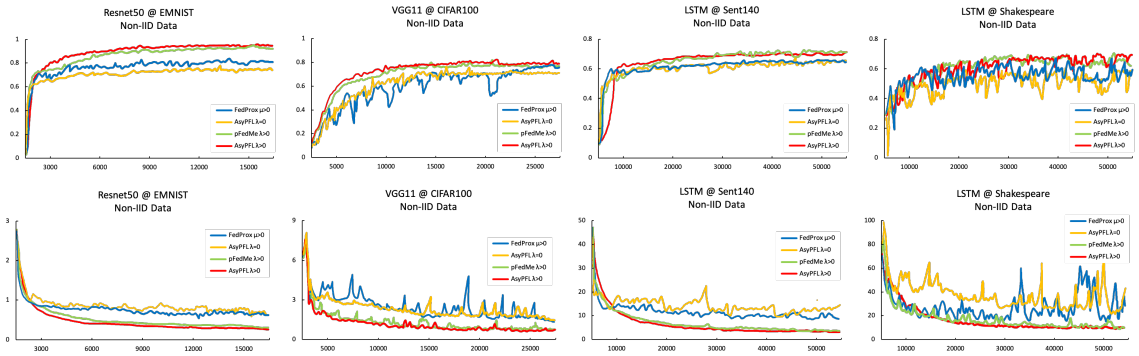

Figure 2: The first row shows the Testing Accuracy-Communication Rounds comparison, and the second row shows the Training Loss-Communication Rounds comparison in non-IID settings. AsyPFL with elastic term stabilizes and improves the convergence of the algorithm.

but they are not stable during training. As this trend can also be observed for Resnet50 in the EMNIST case, it can be concluded that the performance loss originating from the non-IID data is not unique to some functions.

Aiming at better illustrating the effectiveness of the proposed algorithm, we further evaluate and compare AsyPFL with the state-of-the-art algorithms, including FedGATE (Haddadpour et al. (2021)), VRL-SGD (Liang et al. (2019)), APFL (Deng et al. (2020)) and FedAMP (Huang et al. (2021)) on MNIST, CIFAR100, Sentiment140, and Shakespeare dataset. The performance of all the methods is evaluated by the best mean testing accuracy (BMTA) in percentage, where the mean testing accuracy is the average testing accuracy of all participants. For each of the datasets, we apply a non-IID data setting.

Table 3 shows the BMTA of all the methods under the non-IID data setting, which is not easy for vanilla algorithm FedAvg. On the challenging CIFAR100 dataset, VRL-SGD is unstable and performs catastrophically because the models are destroyed, so the customized gradient updates in the method can not tune it up. APFL and FedAMP train personalized models to alleviate the non-IID data. However, the performance of APFL is still damaged by unstable training. FedGATE, FedAMP, and AsyPFL achieve comparably good performance on all datasets.

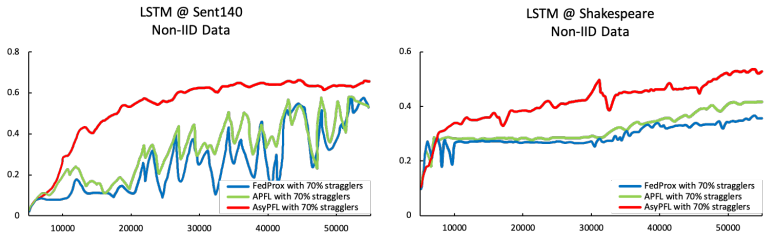

Figure 3: Testing Accuracy-Communication Rounds comparisons among different algorithms with irregular clients. AsyPFL utilizes stale updates from stragglers and is robust to irregular clients.

4.3. Effects of the Elastic Term

AsyPFL utilizes the elastic term scaled by λ, allowing clients to pursue their personalized models in different directions but not to stay far away from the "reference point" ω^*, to which every client contributes. Intuitively, a proper λ restricts the optimization trajectory by limiting the change of the most informative parameters and guarantees convergence.

We explore the impacts of the elastic term by setting different values of λ. Figure 2 shows the performance comparison on different datasets using different models. We compare the result between AsyPFL with $\lambda = 0$ and AsyPFL with best λ. For all datasets, it can be observed that the appropriate λ can increase the stability for unstable methods and can force divergent methods to converge. It also increases accuracy in most cases. As a result, setting $\lambda \geq 0$ is particularly useful in the non-IID setting, which indicates that the AsyPFL benefits practical federated settings.

4.4. Robustness of AsyPFL

Finally, in Figure 3, we demonstrate that AsyPFL is robust to irregular clients. In particular, we track the convergence speed of LSTM trained on Sentiment140 and Shakespeare datasets. It can be observed that a more significant number of irregular clients have adverse effects on all methods. However, the causes for these adverse effects are different: In FedProx and APFL, the actual participation rate is determined by the number of clients that finish the complete training process because it does not include the asynchronous updates. Since irregular clients (stragglers) do not participate in the training, the optimization process can be steered away from the minimum and might even cause catastrophic forgetting. On the other hand, asynchronous updates reduce the convergence speed of AsyPFL by increasing the gradient staleness. The more rounds a client has to wait before being selected to return to training, the more outdated the accumulated gradients become.

5. Conclusion

In this paper, we propose AsyPFL as an asynchronous PFL algorithm that can adapt to heterogeneity issues to improve PFL performance. Our approach uses the elastic term, which helps decompose the personalized model optimization from global model learning, and

allows irregular clients to communicate with the server asynchronously. Theoretical results show that AsyPFL can achieve a state-of-the-art convergence speedup rate. Experimental results demonstrate that AsyPFL outperforms the vanilla PFL algorithms in both convex and non-convex settings, using both IID and non-IID datasets.

Acknowledgments

The work was supported in part by Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, by the National Key Research and Development Program of China under Grant No. 2020AAA0108600, by Shenzhen Outstanding Talents Training Fund 202002, by Guangdong Research Projects No. 2017ZT07X152, No. 2019CX01X104, and No. 2021A1515011825, and by the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001).

References

Pascal Bianchi, Walid Hachem, and Franck Iutzeler. A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. *IEEE Transactions on Automatic Control*, 61(10):2947–2957, 2015.

Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

Tianyi Chen, Georgios B Giannakis, Tao Sun, and Wotao Yin. Lag: Lazily aggregated gradient for communication-efficient distributed learning. *arXiv preprint arXiv:1805.09965*, 2018.

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

Henggang Cui, Hao Zhang, Gregory R Ganger, Phillip B Gibbons, and Eric P Xing. Geeps: Scalable deep learning on distributed gpus with a gpu-specialized parameter server. In *Proceedings of the Eleventh European Conference on Computer Systems*, pages 1–16, 2016.

Wei Dai, Yi Zhou, Nanqing Dong, Hao Zhang, and Eric P Xing. Toward understanding the impact of staleness in distributed machine learning. *arXiv preprint arXiv:1810.03264*, 2018.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

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

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1082–1092. PMLR, 2020a.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020b.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 2350–2358. PMLR, 2021.

Stefan Hadjis, Ce Zhang, Ioannis Mitliagkas, Dan Iter, and Christopher Ré. Omnivore: An optimizer for multi-device deep learning on cpus and gpus. *arXiv preprint arXiv:1606.04487*, 2016.

Ido Hakimi, Saar Barkai, Moshe Gabel, and Assaf Schuster. Taming momentum in a distributed asynchronous environment. *arXiv preprint arXiv:1907.11612*, 2019.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7865–7873, 2021.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. *arXiv: 2012.04221*, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019a.

Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication-efficient local decentralized sgd methods. *arXiv preprint arXiv:1910.09126*, 2019b.

Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. *arXiv preprint arXiv:1506.08272*, 2015.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local SGD with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

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

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

Ioannis Mitliagkas, Ce Zhang, Stefan Hadjis, and Christopher Ré. Asynchrony begets momentum, with an application to deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 997–1004. IEEE, 2016.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *arXiv preprint arXiv:2103.00710*, 2021.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019.

Wei Zhang, Suyog Gupta, Xiangru Lian, and Ji Liu. Staleness-aware async-sgd for distributed deep learning. *arXiv preprint arXiv:1511.05950*, 2015.

Shuxin Zheng, Qi Meng, Taifeng Wang, Wei Chen, Nenghai Yu, Zhi-Ming Ma, and Tie-Yan Liu. Asynchronous stochastic gradient descent with delay compensation. In *International Conference on Machine Learning*, pages 4120–4129. PMLR, 2017.

Pan Zhou, Xiaotong Yuan, Huan Xu, Shuicheng Yan, and Jiashi Feng. Efficient meta learning via minibatch proximal update. *Advances in Neural Information Processing Systems*, 32:1534–1544, 2019.