Probabilistic Node Selection for Federated Learning with Heterogeneous Data in Mobile Edge

Hongda Wu, Ping Wang
Department of Electrical Engineering and Computer Science, York University, Canada
hwu1226@eecs.yorku.ca, pingw@yorku.ca

***Abstract*—Federated Learning (FL) is a distributed learning paradigm that enables a large number of resource-limited nodes to collaboratively train a model without data sharing. The non-independent-and-identically-distributed (non-i.i.d.) data samples invoke discrepancy between the global and local objectives, making the FL model slow to converge. In this paper, we proposed `Optimal Aggregation` algorithm for better aggregation, which finds out the optimal subset of local updates of participating nodes in each global round, by identifying and excluding the adverse local updates via checking the relationship between the local gradient and the global gradient. Then, we proposed a `Probabilistic Node Selection` framework (`FedPNS`) to dynamically change the probability for each node to be selected based on the output of `Optimal Aggregation`. `FedPNS` can preferentially select nodes that propel faster model convergence. Experimental results demonstrate the effectiveness of `FedPNS` in accelerating the FL convergence rate, as compared to `FedAvg` with random node selection.**

***Index Terms*—Federated Learning, Edge Intelligence, Mobile Edge Computing, Node Selection.**

I. Introduction

As the rapid growth of computational capability at mobile edge sides, next-generation computing network is experiencing a paradigm shift from traditional cloud computing to Mobile Edge Computing (MEC) systems [1]. Edge nodes such as sensors, mobile devices, and connected vehicles are generating an unprecedented amount of data consistently, and coupled with cutting-edge Machine Learning (ML) techniques, the MEC system is able to conduct intelligent inference (e.g., road congestion prediction) and perceptive control (e.g., unmanned aerial vehicles (UAVs) swarm navigation [2]). With the ever-increasing computational capability on edge nodes, it becomes more attractive to perform model training on the edge node side instead of sending raw data to the edge server. To this end, Federated Learning (FL) has emerged as a variant of the Distributed ML (DML), which decouples the data acquisition and model training at the edge server [3], facilitates collaboration across nodes, and guarantees privacy.

In general, FL systems aim to optimize a global model under the orchestration of an edge server, which allows the collaboration of multiple edge nodes for data augmentation while keeping training data locally. FL involves several communication rounds, each of which includes local model training, model update transmission, and global model aggregation. Along the iterative process, the edge server is able to train a statistical model that is suitable for all participating nodes without accessing user-sensitive data of edge nodes. The improved data confidentiality and reduced volume of communication cost making FL one of the most promising technologies for future network intelligence [4]. Nonetheless, a fundamental challenge for FL is the data heterogeneity [5], [6]. Specifically, data samples across participating nodes may not be independent and identically distributed (non-i.i.d.). Training on nodes with non-i.i.d. dataset will lead to the biased model update, which stagnates model convergence and reduces the model accuracy, and consequently invokes additional communication rounds to resource-constrained edge nodes [6], [7].

To improve the convergence rate of FL on non-i.i.d. data, a series of studies concentrate on the algorithmic perspective, aiming to reduce communication rounds in FL. These studies include adaptive local training [5], weighting design for model aggregation [7], and node selection [8]–[12]. The algorithm `FedProx` proposed by Li *et al.* [5] uses a regularization term to balance the optimizing discrepancy between global and local objectives and allows participating nodes to perform a variable number of local updates, to consequently overcome the non-i.i.d. data distribution and resource heterogeneity. Authors in [7] exhibited a contribution-related weighting design, namely `FedAdp` to boost the convergence rate of FL with non-i.i.d. data samples, which assigns distinguished weight for participating nodes according to their contribution.

In general, FL algorithms randomly select a subset of nodes (i.e., partial node participation) in each round to participate in local training (e.g., `FedAvg` [3], `FedProx` [5]). Given the data heterogeneity across local nodes, it is not trivial to design node selection schemes that prompt faster convergence. Most of the existing works exploited system heterogeneity and the channel condition [8], [9], [11] to select nodes. Specifically, Nishio *et al.* [8] proposed to select nodes intentionally based on the resource condition on nodes. Amiria *et al.* [9] designed a node scheduling algorithm by considering the significance of local update measured by $\ell 2$ norm and channel condition separately or jointly. Chen *et al.* [10] designed a probabilistic model to make node selection where the probability for each node to be selected is proportional to the norm of local gradient. Similar criteria are adopted in [11], [12], where the significance of local update is evaluated by its gradient divergence. None of the aforementioned node selection designs analyzed the impact of data heterogeneity on node selection.

In this paper, we design a node selection scheme to improve the convergence rate of FL with non-i.i.d nodes, called `FedPNS`, which is a `P`robabilistic `N`ode `S`election

framework with contribution-related criteria. Different from the above probabilistic node selection designs, our work in this paper builds on the data heterogeneity perspective and designs a probabilistic model to choose participating nodes. The proposed method scrutinizes the relationship between local gradients and the global gradient so as to adjust the probability for each node to be selected, which is different from the criteria (i.e., the norm of local gradient/update) adopted in [9]–[12]. Particularly, we find out the global model aggregation over all participating nodes is not of necessity, whereas excluding some adverse local updates may lead to a better global model in terms of training loss. In order to improve the expected decrement of FL loss in each round, we propose an `Optimal Aggregation` algorithm to determine the optimal subset of local updates to be aggregated, which utilizes the inner product between the local gradient and the global gradient[1] as an indicator. By applying the result from `Optimal Aggregation`, the data heterogeneity can be profiled, which is used to adjust the probability for each node to be selected in the subsequent global rounds. Consequently, the server can preferentially select nodes that propel faster model convergence.

II. Preliminaries

In this section, we introduce the key ingredients behind FL, including the system model and the practical algorithm design.

A. Federated Learning Model

In general, federated learning methods [3], [5], are designed to handle the consensus learning task in a decentralized manner, where a central server coordinates the global learning objective and multiple devices train the local model with locally collected data. Consider a network with $\mathcal{K}$ local nodes (i.e., $i \in \{1, 2, \cdots, |\mathcal{K}|\}$), where each node i possesses a local (private) dataset $\mathcal{D}_i$ with size D_i. The nodes are connected with a central server and seek to collaboratively find a global model parameterized by $\mathbf{w}$ that minimizes the empirical risk,

$$F(\mathbf{w}) = \frac{1}{\sum_{i=1}^{|\mathcal{K}|} D_i} \sum_{i=1}^{|\mathcal{K}|} \sum_{\{\mathbf{x},y\}\in\mathcal{D}_i} f(\mathbf{w}, \mathbf{x}, y), \tag{1}$$

where $f(\mathbf{w}, \mathbf{x}, y)$ is the composite loss for training sample $\{\mathbf{x}, y\}$. Specifically, in the context of C-class classification problem hereinafter, each training sample $\{\mathbf{x}, y\} \in \mathcal{D}_i$ is assumed to contain a feature vector $\mathbf{x}$ and label y over feature space $\mathbb{X}$ and label space $\mathbb{Y}$ (i.e., $\mathbb{Y} = [C]$, where $[C] = \{1, \cdots, C\}$). For each available training sample $\{\mathbf{x}, y\} \in \bigcup_i \mathcal{D}_i$ in the FL problem, the FL model parameterized by $\mathbf{w}$ is considered to learn the predicted probability vector $\bar{\mathbf{y}}$, i.e., $\bar{\mathbf{y}} | \sum_{j=1}^{C} \bar{y}_j = 1, \bar{y}_j \geq 0, \forall j \in [C]$, with empirical risk. From a federation perspective, the global objective $F(\mathbf{w})$ in (1) is surrogated by local objective $F_i(\mathbf{w})$ and can be represented as

$$F(\mathbf{w}) = \sum_{i=1}^{|\mathcal{K}|} \frac{D_i}{\sum_{i=1}^{|\mathcal{K}|} D_i} F_i(\mathbf{w}), \tag{2}$$

[1]We use local/global gradient and local/global update interchangeably.

For each node i, $F_i(\mathbf{w})$ commonly measures the empirical risk (e.g., cross entropy loss) over the dataset $\mathcal{D}_i$ with possibly different data distribution $q^{(i)}$, which is defined as follows

$$\begin{aligned} F_i(\mathbf{w}) &= \mathbb{E}_{\mathbf{x},y\sim q^{(i)}} \left[-\sum_{j=1}^{C} \mathbb{1}_{y=j} \log l_j(\mathbf{w}, \mathbf{x}, y) \right] \\ &= -\sum_{j=1}^{C} q^{(i)}(y=j) \mathbb{E}_{\mathbf{x}|y=j} \left[\log l_j(\mathbf{w}, \mathbf{x}, y) \right], \end{aligned} \tag{3}$$

where $l_j(\mathbf{w}, \mathbf{x}, y)$ denotes the probability that the data sample $\{\mathbf{x}, y\}$ is classified as the j-th class given model $\mathbf{w}$. $q^{(i)}(y = j)$ denotes the data distribution on node i over class $j \in [C]$.

B. FedAvg with Partial Node Participation

The most commonly used algorithm to solve (2) is Federated Averaging (`FedAvg`) [3], where the training consists of multiple communication rounds. At each communication round t, the server selectes a fraction c of nodes $|\mathcal{S}_t| = c|\mathcal{K}|$ to participate in the training. Taking the global model $\mathbf{w}^{t-1}$ in previous round as the reference, each participating node $i \in \mathcal{S}_t$ performs τ steps of local Stochastic Gradient Descent (SGD) to optimize its objective

$$\mathbf{w}_i^t = \mathbf{w}^{t-1} - \eta \nabla F_i(\mathbf{w}^{t-1}), \tag{4}$$

where η is the learning rate and $\nabla F_i(\cdot)$ is the gradient[2] at node i. (4) gives a general principle of SGD optimization, where $\mathbf{w}_i^t$ is the result after τ local updates of mini-batch SGD (i.e., $\tau = \frac{D_i}{B} E$, where E is the number local training epochs, B is the batch size of mini-batch training samples).

The participating nodes then communicate their model update $\Delta_i^t = \mathbf{w}_i^t - \mathbf{w}^{t-1}$ back to the server, which aggregates them and updates the global model[3] as follows

$$\begin{aligned} \Delta^t &= \frac{1}{|\mathcal{S}_t|} \sum_{i\in\mathcal{S}_t} \Delta_i^t \\ \mathbf{w}^t &= \mathbf{w}^{t-1} + \Delta^t. \end{aligned} \tag{5}$$

Though `FedAvg` can achieve a decent convergence rate with trivial node selection policy and simple averaging design, model performance on non-i.i.d. dataset is not satisfactory [6], [7]. In trivial node selection policy (e.g., random selection in `FedAvg`), the distribution of data samples on selected nodes $q^{(i)}$ differs from each other. Local updates lead the model towards optima to its local objective, which may deviate from the global objective in a non-i.i.d. setting, causing training instability that makes the FL model struggle to converge. Therefore, it is crucial to analyze the node selection policy from the data heterogeneity perspective; identifying and choosing the nodes that contribute better to model convergence.

[2]Through this paper, the gradient refers to the stochastic version instead of the actual gradient calculated from the entire dataset.

[3]It is worth mentioning that the aggregation scheme is applied over all nodes in the vanilla `FedAvg` [3], i.e., $\Delta^t = \sum_{i\in\mathcal{S}_t} \psi_i \Delta_i^t + \sum_{i\in\mathcal{K}-\mathcal{S}_t} \psi_i \mathbf{w}^{t-1}$, where $\psi_i = D_i / (\sum_{i=1}^{|\mathcal{K}|} D_i)$. The subsequent work [5] proposed a variant of aggregation over participating nodes as in (5). Hereinafter, `FedAvg` denotes the algorithm that involves random selection and partial aggregation of nodes with equal data size [5].

III. Contribution-based Node Selection

In this section, we design a probabilistic node selection scheme to improve the convergence rate of federated learning. For FL with the heterogeneous dataset, we analyze the convergence property of `FedAvg` theoretically (Section III-A). In Section III-B, we challenge the necessity of the global model aggregation over all participating nodes. Then, the `Optimal Aggregation` algorithm is proposed, which can exclude the adversarial local updates to make greater progress in reducing the expected decrement of global loss in each round. Finally, the FL with Probabilistic Node Selection (`FedPNS`) is proposed based on the result of `Optimal Aggregation`.

A. Convergence Analysis

For theoretical analysis purposes, we employ the following assumptions to the loss function, which have also been commonly made in the literature [5], [13], [14].

Assumption 1. ***L*-Lipschitz smooth**.
$F_i(\mathbf{w})$ is L-Lipschitz smooth for all node i.

Assumption 2. ***δ*-local dissimilarity**.
Local loss functions $F_i(\mathbf{w}^t)$ are δ-local dissimilar at $\mathbf{w}^t$, i.e., $\mathbb{E}_{i\backsim\mathcal{S}_t}\left[\|\nabla F_i(\mathbf{w}^t)\|^2\right] \leq \|\nabla F(\mathbf{w}^t)\|^2\delta^2$ for $i \in \mathcal{S}_t$ and $t = 1,\cdots,T$, where T is the number of global rounds. $\mathbb{E}_{i\backsim\mathcal{S}_t}[\cdot]$ denotes the expectation over participating nodes $\mathcal{S}_t$ with weight $\frac{1}{|\mathcal{S}_t|}$ (as in (5)). $\nabla F(\mathbf{w}^t)$ is the global gradient at the t-th global round defined as $\nabla F(\mathbf{w}^t) = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\nabla F_i(\mathbf{w}^t)$.

Theorem 1. *Let assumptions 1 and 2 hold. Suppose that $\mathbf{w}^t$ is not a stationary solution, the expected decrement on the global loss of FedAvg between two consecutive rounds satisfies*

$$F(\mathbf{w}^{t+1}) \leq F(\mathbf{w}^t) - \eta\mathbb{E}_{i\backsim\mathcal{S}_t}\left[\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right] + \frac{L\eta^2}{2}\|\nabla F(\mathbf{w}^t)\|^2\delta^2, \quad (6)$$

where $\langle\cdot\rangle$ is the inner product operation, and $\|\cdot\|$ denotes the $\ell 2$ norm of a vector.

The proof of Theorem 1 is omitted due to the page limit. Theorem 1 provides a bound on how rapid the decrease of the global FL loss can be expected. The decrease of global FL loss between two consecutive rounds shows a dependency on δ, which represents the variance between local data distributions, and the aggregation strategy $\mathbb{E}_{i\backsim\mathcal{S}_t}[\cdot]$, where $\nabla F(\mathbf{w}^t)$ is obtained by aggregating over local updates from all participating nodes, i.e., $\nabla F_i(\mathbf{w}^t), i \in \mathcal{S}_t$ with weight $1/|\mathcal{S}_t|$.

B. Aggregation with Gradient Information

In the vanilla `FedAvg` [3] and the subsequent work [5], [6], the averaging technique is used for global update aggregation due to its simplicity. One can challenge the inherent rule that the global update is aggregated over local updates of all participating nodes since the local updates may contribute global model in an adverse way. As a sanity check, at any communication round t, the local update from the participating nodes whose inner product between their gradients and global gradient is negative i.e., $\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle < 0$, will slow the model convergence because of the reduced expected loss decrement (i.e., a lower expectation value as in (6)) in this round. As such, it is not trivial to exclude the adverse local updates, which is realized by examining the value of expectation term in Theorem 1, as illustrated later. Excluding adverse local updates gives an impact on the reduction of overall data heterogeneity, thus changes the relationship between the local gradient and the global gradient $\langle\nabla\bar{F}(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle$, where $\nabla\bar{F}(\mathbf{w}^t) = \frac{1}{|\bar{\mathcal{S}}_t|}\sum_{i\in\mathcal{S}_t^*}\nabla F_i(\mathbf{w}^t)$ is defined over $\mathcal{S}_t^*$, i.e., the subset of participating nodes $\mathcal{S}_t$ after successfully excluding the nodes with adverse local updates.

To find the optimal subset of local updates to aggregate, we first check the expectation term $\mathbb{E}_{i\backsim\mathcal{S}_t}\left[\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right]$ in Theorem 1 and exclude the local updates from participating nodes k, i.e., $k \in \mathcal{S}_t - \bar{\mathcal{S}}_t$ if $\mathbb{E}_{i\backsim\bar{\mathcal{S}}_t}\left[\langle\nabla\bar{F}(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right] > \mathbb{E}_{i\backsim\mathcal{S}_t}\left[\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right]$ is satisfied. However, excluding local updates gives an impact on the global update and overall data heterogeneity, i.e., $\|\nabla F(\mathbf{w}^t)\|^2\delta^2$, the last term on the right hand side of (6), which makes the expected decrement of global loss, i.e., $\Delta F(\mathbf{w}^t) = \frac{L\eta^2}{2}\|\nabla F(\mathbf{w}^t)\|^2\delta^2 - \eta\mathbb{E}_{i\backsim\mathcal{S}_t}\left[\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right]$, difficult to be analyzed quantitatively given L and δ. Therefore, in the second step, test loss is adopted to ensure that excluding local updates makes global update better in terms of model convergence. In particular, the global model $\mathbf{w}^{t+1}$ and $\bar{\mathbf{w}}^{t+1}$ generated by $\nabla F_i(\mathbf{w}^t), i \in \mathcal{S}_t$ and $\nabla F_i(\mathbf{w}^t), i \in \bar{\mathcal{S}}_t$, respectively, are evaluated using mini-batch of samples with size $\bar{B}$ that are sampled uniformly at random from $\mathcal{D}_{test}$ (e.g., test dataset in MNIST).

An iterative algorithm called `Optimal Aggregation` is proposed for a better local update aggregation in each round, which finds the *optimal* subset of local update $\Delta_i, i \in \mathcal{S}_t^* \subseteq \mathcal{S}_t$ by excluding the adverse local updates Δ_k, $k \in \mathcal{S}_t - \mathcal{S}_t^*$, as in Algorithms 1. Specifically, for a given set of participating nodes $\mathcal{S}_t$ in each global round t, the server iteratively removes one of the local updates $\nabla F_i(\mathbf{w}^t), i \in \mathcal{S}_t$, generates the potential global gradient, and calculates the expectation term in (6) (i.e., Check Expectation, line 18-21). If excluding one local update gives a higher expectation value, compared with the case that includes all local updates retained in $\mathcal{S}_t$, that local update will be labeled, and loss comparison will be performed to check the loss criterion (Check Loss, line 22-25), otherwise the server keeps all local updates (line 6). If the loss criterion is satisfied (line 13), the labeled local update is eventually removed from set $\mathcal{S}_t$ (line 14). Otherwise, the server keeps that local update retained in $\mathcal{S}_t$ (line 12). The process repeats until no adverse local update can be found or the number of remaining local updates is below a threshold v (line 4). In Algorithm 1, the function `pop` is defined as removing element (line 14). The introduced "temp" is a dictionary with key-value pairs (line 5) and the function `max` returns the maximum value (line 6) or the key (i.e., the node index i) corresponding to that value (line 9), respectively.

Given a set of participating nodes $\mathcal{S}_t$, the benefits of finding optimal local updates are twofold: (i) Excluding the potential local updates that contribute to the global model adversely

Algorithm 1 Optimal Local Updates for Aggregation

Procedure OPTIMAL AGGREGATION
Input: $\mathcal{S}_t$, Δ_i^t, v, temp = {}
$\nabla F(\mathbf{w}_i^t) = -\Delta_i^t/\eta$
$\nabla F(\mathbf{w}^t) = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t}\nabla F_i(\mathbf{w}^t)$
max = $\mathbb{E}_{i\sim\mathcal{S}_t}\left[\langle\nabla F(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right]$
while $|\mathcal{S}_t| \geq v$ **do**
  temp $\leftarrow$ CHECK EXPECTATION ($\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t$, temp)
  if max(temp).value < max **do**
    break with $\mathcal{S}_t^* = \mathcal{S}_t$
  else
    key = max(temp).key
    ls($\mathbf{w}$), ls($\bar{\mathbf{w}}$), $\bar{\mathcal{S}}_t \leftarrow$ CHECK LOSS ($\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t$, key)
    if ls($\mathbf{w}$) > ls($\bar{\mathbf{w}}$) **do**
      break with $\bar{\mathcal{S}}_t$, $\mathcal{S}_t^* = \mathcal{S}_t$
    else
      $\mathcal{S}_t, \mathcal{S}_t^* \leftarrow \mathcal{S}_t$.pop(key)
      max $\leftarrow$ temp(key).value
return $\mathcal{S}_t^*$, $\bar{\mathcal{S}}_t$
$\mathbf{w}^{t+1} \leftarrow$ GLOBAL UPDATE ($\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t^*$)
Procedure CHECK EXPECTATION
Input: $\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t$, temp
for $i = 1, \cdots, |\mathcal{S}_t|$ **do**
  $\bar{\mathcal{S}}_t \leftarrow \mathcal{S}_t$.pop($\mathcal{S}_t[i]$)
  $\nabla\bar{F}(\mathbf{w}^t) = \frac{1}{|\bar{\mathcal{S}}_t|}\sum_{i\in\bar{\mathcal{S}}_t}\nabla F_i(\mathbf{w}^t)$
  temp($\mathcal{S}_t[i]$) = $\mathbb{E}_{i\sim\bar{\mathcal{S}}_t}\left[\langle\nabla\bar{F}(\mathbf{w}^t), \nabla F_i(\mathbf{w}^t)\rangle\right]$
Procedure CHECK LOSS
Input: $\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t$, key
$\bar{\mathcal{S}}_t \leftarrow \mathcal{S}_t$.pop(key)
Generate global model $\mathbf{w}^{t+1}$ by $\nabla F_i(\mathbf{w}^t), i \in \mathcal{S}_t$ and $\bar{\mathbf{w}}^{t+1}$ by $\nabla F_i(\mathbf{w}^t), i \in \bar{\mathcal{S}}_t$, respectively
Evaluate $\mathbf{w}^{t+1}$, $\bar{\mathbf{w}}^{t+1}$ by using mini-batch samples from $\mathcal{D}_{test}$ and get the loss ls($\mathbf{w}$) and ls($\bar{\mathbf{w}}$), respectively
return ls($\mathbf{w}$), ls($\bar{\mathbf{w}}$), $\bar{\mathcal{S}}_t$
Procedure GLOBAL UPDATE
Input: $\nabla F_i(\mathbf{w}^t)$, $\mathcal{S}_t^*$
Generate $\mathbf{w}^{t+1}$ by $\nabla F_i(\mathbf{w}^t), i \in \mathcal{S}_t^*$ via (4) and (5)
return $\mathbf{w}^{t+1}$

results in a larger decrement of the expected loss in each round. (ii) By CHECK EXPECTATION, the potential adversarial nodes $k, k \in \mathcal{S}_t - \bar{\mathcal{S}}_t$ (nodes with non-i.i.d. dataset normally) are identified. This identification can be used for consequent probabilistic node selection, as illustrated in Section III-C.

C. Probabilistic Node Selection

Under the context of probabilistic node selection, it is natural to lower the node selection probabilities for those nodes whose local updates slow model convergence. Therefore, on the server-side, we propose to dynamically change the probability for each node to be selected via using the output of `Optimal Aggregation` (i.e., $\bar{\mathcal{S}}_t$). In particular, the probabilities for those nodes that are labeled by the procedure CHECK EXPECTATION (i.e., $i \in \mathcal{S}_t - \bar{\mathcal{S}}_t$) are decreased according to the parameter x in (7), and the probabilities for all the rest nodes will be increased.

$$\Delta p_i^t = p_i^t \cdot \min[(x+\beta)^\alpha,\ 1], \quad i \in \mathcal{S}_t - \bar{\mathcal{S}}_t, \tag{7}$$

where p_i^t and Δp_i^t denote the probability for node i to be selected in the t-th global round, and its probability decrement in next round, respectively. `min` function returns the minimum value among all arguments, $x \in (0, 1]$ is defined as the ratio between the accumulated times that a node is labeled by the procedure CHECK EXPECTATION and the accumulated times that the node is selected, $\alpha \in \mathbb{Z}^+, \beta \in [0, 1]$ are coefficients as explained in the following.

- $\lim_{x\to\epsilon} x^\alpha + \beta \approx 1$, where $\epsilon \propto \alpha$ is constant.
- $\lim_{0\to x\to\upsilon} x^\alpha + \beta \approx \beta$, where $\upsilon \propto \alpha$ is a constant.

α controls how big the probability decrement is achieved by $(x+\beta)^\alpha$ given a ratio x. For example, a large value of α brings an aggressive decrement since the probability decrement happens in a wide range $(\beta, 1)$ as x increases within a small range (υ, ϵ), making the probability drop very quickly when x grows. Meanwhile, the large α makes node selection sensitive to the identification mistake, which may prevent i.i.d. nodes from being selected in the subsequent rounds. However, setting a small value of α is not consistently effective to differentiate the nodes since the probability change is marginal. β is adopted to keep the rate of probability change in a visible range $[\beta, 1]$. From experiments, we find out $\alpha = 2, \beta = 0.7$ is a good choice that balances the tradeoff.

After getting the probability change for the labeled nodes (i.e., $i \in \mathcal{S}_t - \bar{\mathcal{S}}_t$), we equally increase the probability for all the rest nodes $i \in \mathcal{K} - (\mathcal{S}_t - \bar{\mathcal{S}}_t)$, as shown in (8).

$$p_i^{t+1} = \begin{cases} p_i^t - \Delta p_i^t & i \in \mathcal{S}_t - \bar{\mathcal{S}}_t \\ p_i^t + \frac{\sum_{i\in\mathcal{S}_t-\bar{\mathcal{S}}_t}\Delta p_i^t}{|\mathcal{K}-(\mathcal{S}_t-\bar{\mathcal{S}}_t)|} & i \in \mathcal{K} - (\mathcal{S}_t - \bar{\mathcal{S}}_t) \end{cases}, \tag{8}$$

where $p_i^{t+1}, i \in \mathcal{K}$ are used for the $(t+1)$-th round.

The proposed FL design with probabilistic node selection and optimal aggregation is summarized in Algorithm 2.

Algorithm 2 FL with Probabilistic Node Selection

Procedure FEDERATED OPTIMIZATION
Input: $E, B, \eta, \mathcal{K}, T, p_i^t\ i = 1, \cdots, |\mathcal{K}|$
Server initializes $\mathbf{w}^0, p_i^0 = 1/|\mathcal{K}|$
for $t = 1, \cdots, T$ **do**
  Server samples a subset $\mathcal{S}_t$ of nodes according to p_i^{t-1}
  Server sends $\mathbf{w}^t$ to nodes $i \in \mathcal{S}_t$
  Each node $i \in \mathcal{S}_t$ finds $\mathbf{w}_i^t$ to optimize $F_i(\mathbf{w}^t)$ using SGD, as in (4), and sends back Δ_i^t to the server
  $\mathbf{w}^{t+1}$, $\bar{\mathcal{S}}_t \leftarrow$ OPTIMAL AGGREGATION
  Server updates the probability $p_i^t\ i = 1, \cdots, |\mathcal{K}|$ by (7) and (8) for next round's usage
return $\mathbf{w}^T$
Procedure OPTIMAL AGGREGATION
Input: $\mathcal{S}_t$, Δ_i^t, v, temp = {}
Direct to Algorithm 1
return $\mathbf{w}^{t+1}$, $\bar{\mathcal{S}}_t$

IV. EVALUATION AND ANALYSIS

We now present empirical results for the proposed probabilistic node selection strategy on image classification task using MNIST and CIFAR-10 dataset with different learning objectives. Meanwhile, the commonly used benchmark `FedAvg` is adopted as comparison. We first demonstrate the effectiveness of the proposed `Optimal Aggregation` in

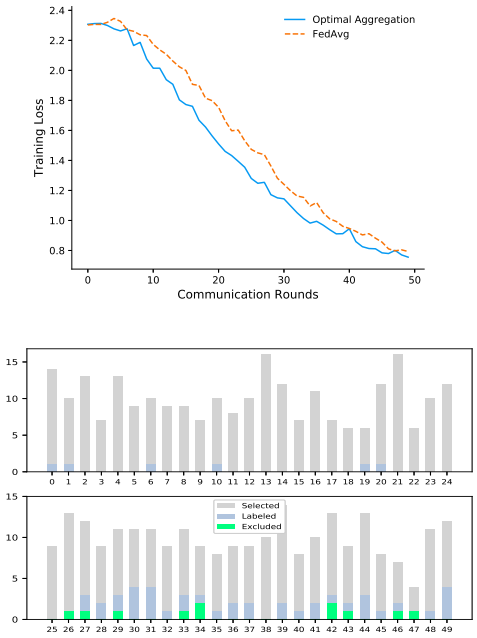

Fig. 1. Performance of proposed `Optimal Aggregation`. (1) Upper: The training loss on the MNIST dataset when different aggregation strategies are adopted. `Optimal Aggregation` and `FedAvg` aggregate local updates over $\mathcal{S}_t^*$ and $\mathcal{S}_t$, respectively. (2) Bottom: We use a triple to observe the result of `Optimal Aggregation`. The upper and bottom row refer to the results for i.i.d. nodes and non-i.i.d. nodes, respectively.

enlarging the expected decrement of FL global loss and in identifying the potential adversarial nodes (Section IV-A). Then, the superiority of the proposed `FedPNS` in the presence of various data heterogeneity is illustrated in Section IV-B.

Through the experimental result, unless otherwise specified, we evaluate the accuracy of the trained models using the testing set from each dataset. The fraction for selecting nodes is set to be $c = 0.2, |\mathcal{S}_t| = c|\mathcal{K}| = 10$, $D_i = 200$, $B = 20$, $E = 1$, $T = 200$, $\eta = 0.01$, decay rate $= 0.995$, $v = 0.7$, $\bar{B} = 128$. The overall data heterogeneity is measured by σ and the skewness of dataset on non-i.i.d. nodes is represented by ρ. For example, $\sigma = 0.2, \rho = 2$ means that $\sigma|\mathcal{K}| = 10$ nodes are equipped with i.i.d. dataset, where non-i.i.d. dataset lay on the rest $(1-\sigma)|\mathcal{K}| = 40$ nodes, and the data samples on which are evenly belong to 2 labels. As such, a smaller value of σ and ρ indicates a higher data heterogeneity.

A. Performance of `Optimal Aggregation`

We conduct an experiment to illustrate the performance of the proposed `Optimal Aggregation` algorithm. Particularly, we train a CNN model[4] on MNIST dataset. The data heterogeneity is set to be $\sigma = 0.5, \rho = 1$. In each global round, we randomly select $|\mathcal{S}_t| = 10$ nodes while guaranteeing the participating nodes include half i.i.d. nodes and half non-i.i.d. nodes. To avoid the randomness of node selection, the participating nodes in each round are kept as the same for `FedAvg` [5] and the proposed Algorithm 1.

As shown in the upper part of Fig. 1, the proposed `Optimal Aggregation` algorithm can achieve lower training loss than `FedAvg`. When the global model is not robust in several initial rounds, the local updates are more diverse due to the data heterogeneity, thus excluding adverse local updates is more effective. We count the accumulated times that each node is selected, labeled by the procedure CHECK EXPECTATION (line 7 in Algorithm 1), and finally excluded by the procedure CHECK LOSS (line 14 in Algorithm 1). As we can see from the bottom part of Fig. 1, i) the i.i.d nodes (i.e., with index "0", $\cdots$, "24") are never excluded, yet some of the non-i.i.d nodes (e.g.,"26", "27", "34", etc.) have been excluded many times. ii) Almost all non-i.i.d. nodes were labeled at least one time, which illustrates the effectiveness of `Optimal Aggregation` in identifying the nodes with the skewed dataset.

B. Comparison between `FedPNS` and `FedAvg`

In this section, we use different combinations of σ and ρ to investigate the performance of the proposed `FedPNS` scheme in the presence of different data heterogeneity. The Multinomial Logistic Regression (MLR) model and CNN model are adopted to represent convex and non-convex learning objectives, respectively. Through all experiments, α and β are chosen to be 2 and 0.7 respectively. The number of communication rounds T is set to be 100 for MLR.

As we can tell from Fig. 2, `FedPNS` converges faster and achieves a higher test accuracy, compared with `FedAvg` for both MLR and CNN models regardless of different data heterogeneity. `FedPNS` achieves better improvement when the CNN model is adopted, compared with the scenario when the MLR model is utilized, which attributes to the limited learning capability of MLR. In addition, it is observable that as the data becomes more heterogeneous, the performance enhancement is enlarged (i.e., α decreases from 0.5 to 0.2 for a given β, or β changes from 2 to 1 for a given α). When the number of i.i.d. nodes is limited and the non-i.i.d nodes are equipped with highly skewed dataset (e.g., $\sigma = 0.2, \rho = 1$ and $\sigma = 0.3, \rho = 1$), `FedPNS` gains remarkable performance improvement, which verifies the effectiveness of `FedPNS` in identifying and selecting the nodes that contribute global model better. For the scenario with the lowest data heterogeneity (i.e., $\sigma = 0.5, \rho = 2$), the performance gap between `FedPNS` and `FedAvg` is not obvious. This is because the impact of the non-i.i.d. nodes on the convergence is reduced when a large number of i.i.d. nodes can be selected.

For the more complex three channel image classification task over CIFAR-10 dataset, the number of local epoch is set to be $E = 5$. As we can see from the lower subplots of Fig. 2, compared with `FedAvg`, `FedPNS` converges faster and leads to a higher test accuracy, especially for the high data heterogeneity scenario (i.e., $\sigma = 0.2$ and $0.3, \rho = 1$). The

[4]The CNN model has 7 layers with the following structure: $5 \times 5 \times 10$ Convolutional $\rightarrow 2 \times 2$ MaxPool $\rightarrow 5 \times 5 \times 20$ Convolutional (50% dropout) $\rightarrow 2 \times 2$ MaxPool $\rightarrow 320 \times 50$ Fully connected $\rightarrow 50 \times 10$ Fully connected $\rightarrow$ Softmax. ReLu activation applies to convolutional and fully connected layers.

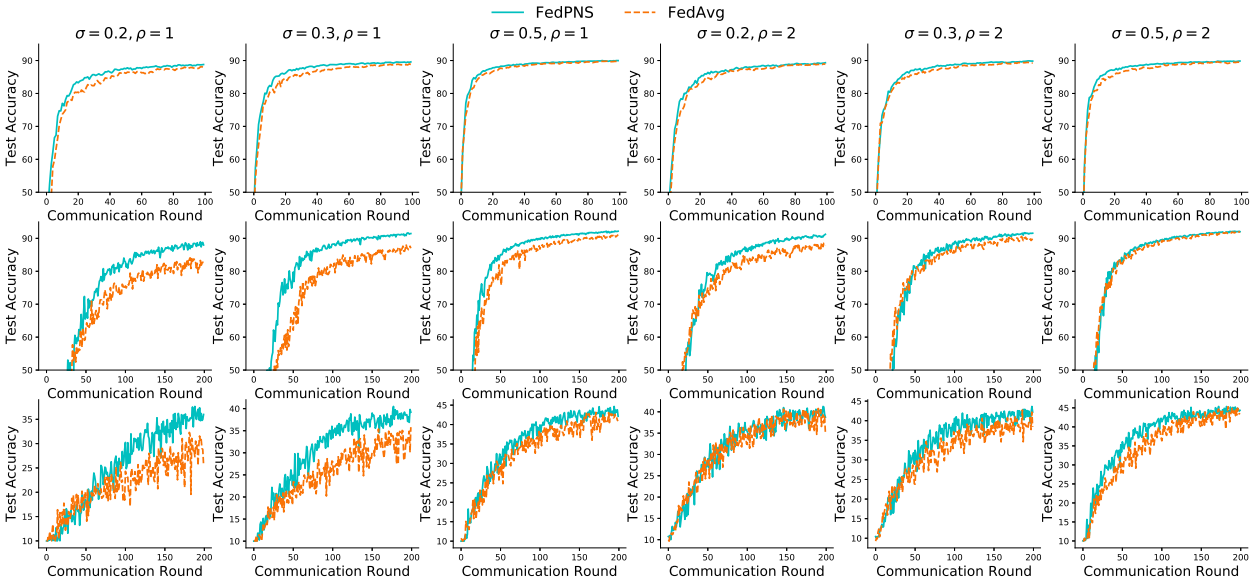

Fig. 2. Test accuracy over communcation rounds of `FedPNS` and `FedAvg` with different data heterogeneity. Upper and middle subplots correspond to training performance on MNIST dataset when the MLR model and CNN model are adopted, respectively. The lower subplots show the result on CIFAR-10 dataset.

performance improvement of `FedPNS` is not obvious when $\sigma = 0.2, \rho = 2$, this is because the small number of i.i.d. nodes with less heterogeneous data samples on non-i.i.d. nodes makes `FedPNS` hard to distinguish the node contribution.

V. Conclusion

In this paper, we have presented our design of `FedPNS` algorithm, a probabilistic node selection strategy that can preferentially select nodes to boost model convergence of FL with non-i.i.d. datasets. `FedPNS` adjusts the probability for each node to be selected in each round based on the result of the proposed `Optimal Aggregation` algorithm, which is able to find out the optimal subset of local updates from participating nodes and excludes the adverse local updates for a better model aggregation, by measuring the relationship between the local gradient and the global gradient from participating nodes. Experimental results have shown that FL training with `FedPNS` accelerates model convergences and leads to higher test accuracy over the widely adopted MNIST and CIFAR-10 datasets, as compared to `FedAvg`.

References

[1] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. the Artificial Intelligence and Statistics Conference (AISTATS)*, 2017.

[4] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.

[5] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. of Machine Learning and Systems (MLSys)*, 2020.

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

[7] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Transactions on Cognitive Communications and Networking*, Early Access, 2021.

[8] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. the IEEE International Conference on Communications (ICC)*, 2019.

[9] M. M. Amiri, D. Gndz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3643–3658, 2021.

[10] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[11] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7690–7703, 2020.

[12] E. Rizk, S. Vlaski, and A. H. Sayed, "Optimal importance sampling for federated learning," in *Proc. the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[13] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2020.

[14] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. the International Conference on Learning Representations (ICLR)*, 2020.