Local Performance Trade-Off in Heterogeneous Federated Learning with Dynamic Client Grouping

Yingchi Mao\textsuperscript{a,b}, Jun Wu\textsuperscript{b}, Yangkun Cheng\textsuperscript{b}, Ping Ping\textsuperscript{a,b}, and Jie Wu\textsuperscript{c}

\textsuperscript{a} Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
\textsuperscript{b} School of Computer and Information, Hohai University, Nanjing, China
\textsuperscript{c} Center of Networked Computing, Temple University, Philadelphia, USA

IEEE MASS
The 19\textsuperscript{th} IEEE International Conference on Mobile Ad-Hoc and Smart Systems
Oct. 20, 2022
1. Introduction
2. Related Work
3. Approach
4. Experiment
5. Conclusion
Introduction

Background

Although client selection can coordinate large-scale clients for efficient training, the inherent characteristic of the scheme, i.e., the partial client participation can lead to a performance bias of the global model among clients.
■ Optimization object

Improve *unbalanced local performances* on different clients of the global model caused by client selection

■ Goal

Balance the client participation in local training while guaranteeing that clients are not discarded.
Related Work

**Client Selection**
- Drop partial clients
- Drop local models
- Drop local parameters

**Performance biases**

**Client Grouping**
- Local training time
- Local data samples
- Keep similar clients within the group
Related Work

Client Grouping – local training time

- Group clients according to local training time
- Clients with long training time due to large data volume are despised
Related Work

Client Grouping – local data samples

- Group clients according to data volume
- Clients with large data volume seriously **slow down** the training process
Approach

Federated Learning Clients Dynamic Grouping Method based on Local Computational Efficiency

1. Initialize
2. Construct polynomial distribution set
3. Calculate local computational efficiency
4. Group clients
5. Update local computational efficiency

- Calculate the probability of a client sampled in each polynomial distribution
- The client belongs to the group corresponding to the distribution with the highest sampling probability
- Randomly select clients from each group
- Subsets of clients involved in training
- Update the local computational efficiency of corresponding clients after training
- Local computational efficiency $E = \{e_i \mid i = 1, \ldots, n\}$
Approach

Calculate local computational efficiency

\[ e_i = \begin{cases} d_i, & c_i \left( r_{\text{latest}} = 0 \right) \\ d_i, & t_i^{\text{latest}}, c_i r_{\text{latest}} \left( r_{\text{latest}} \neq 0 \right), i \in \{1, \ldots, n\} \end{cases} \]
Approach

Construct Polynomial Distribution Set

A polynomial distribution set of length m

\[ \sum_{i=1}^{m} m \varepsilon_i = mM \]

\[ m \varepsilon_{i-1} \quad m \varepsilon_i \quad m \varepsilon_{i+1} \]

\[ \cdots \quad D_{k-1} \quad D_k \quad D_{k+1} \quad \cdots \]

\[ q_{k-1,j-1} \quad q_{j-1} \quad q_{k,j} \quad q_{k,j+1} \quad q_{k+1,j+1} \]

\[ M \quad M \quad M \]

The priority of assigning clients’ local computation efficiency is offered to the polynomial distribution

\[ \forall k \in \{1, \ldots, m\}, \sum_{i=1}^{n} q_{i,k}^r = 1, q_{i,k}^r \geq 0 \]

Sampling probability
Approach

Group Clients

Algorithm 1 Client Grouping

Input: \( n \): number of clients, \( m \): number of client groups, \( r \): current iteration round, \( r_u \): iteration round scheduled for updating client grouping results, \( \{e_i \mid i = 1,...,n\} \): client local computational efficiency set

Output: \( \{q_{i,k}'\mid i = 1,...,n; k = 1,...,m\} \): probability that a client belongs to each client group

1: if \( r \% r_u == 0 \) then
2: define \( k = 1 \)
3: define \( count = 0 \)
4: define \( M = \sum e_i \)
5: for \( i = 1 \) to \( n \) do
6: \( count = count + me_i \)
7: \( count = M\alpha_i + \beta_i \)
8: if \( \alpha_i > k \) then
9: \( (q_{i,k}'') = M - \beta_{i-1} \)
10: \( \forall l \geq k + 1 s.t. (\alpha_i - 1) - l \geq 0, (q_{i,k}') = M \)
11: end if
12: \( (q_{i,k}')^Y = \beta_i \)
13: \( k = \alpha_i \)
14: end for
15: return \( \{q_{i,k}' = \frac{(q_{i,k}')^Y}{M} \mid i = 1,...,n; k = 1,...,m\} \)
16: end if

Build a set of polynomial distributions for each client group

Output the probability that a client belongs to each polynomial distribution (Client Grouping)
Approach

**Update Local Computational Efficiency**

**Algorithm 2 Local Computational Efficiency Update**

- **Input:** \( \tau \): current iteration round, \( \{d_i \}_{i=1,...,n} \): number of local data on each client, \( S^\tau \): subset of clients involved in training in iteration round \( \tau \)
- **Output:** \( \{e_i \}_{i=1,...,n} \): Local computational efficiency set of each client

```
1: define \( E = \{e_i = 0 \}_{i = 1,...,n} \)
2: if \( \tau = 1 \) then
3:   for \( i = 1 \) to \( n \) do
4:     \( e_i = d_i \)
5:   end for
6: else
7:   \( S^\tau = \text{ClientGrouping}(E) \)
8:   \( \{t^\tau_i | e_i \in S^\tau \} = \text{Train()} \)
9:   for \( c_i \in S^\tau \) do
10:      \( e_i = |d_i|/t^\tau_i \)
11:   end for
12:   \text{Descend}(E)
13: end if
14: return \( E = \{e_i | i = 1,...,n \} \)
```

- Initialize clients’ local computational efficiency using their local data quantity
- Update the local computational efficiency of clients participating in training
The clients and server are both on the **same LAN** and communicate each other via the **PySyft’s WebSocket protocol**.
Two different CNN models are chosen as experimental models for the MNIST-Fed dataset and the CIFAR-10-Fed dataset.
### Analysis of Hyperparameter Selection for FedGLCE

#### TABLE III  
**Experimental results of FedGLCE in different group update periods (MNIST-Fed, 5 groups)**

<table>
<thead>
<tr>
<th>Group update period $r_u$</th>
<th>Variance of participation</th>
<th>Variance of local accuracy</th>
<th>Accuracy of global test</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>19.54</td>
<td>262.48</td>
<td>81.82%</td>
</tr>
<tr>
<td>10</td>
<td>17.92</td>
<td>61.41</td>
<td>86.99%</td>
</tr>
<tr>
<td>20</td>
<td><strong>16.54</strong></td>
<td><strong>37.47</strong></td>
<td><strong>91.14%</strong></td>
</tr>
<tr>
<td>50</td>
<td>11.76</td>
<td>80.41</td>
<td>90.74%</td>
</tr>
</tbody>
</table>

#### TABLE IV  
**Experimental results of FedGLCE in different group update periods (MNIST-Fed, 10 groups)**

<table>
<thead>
<tr>
<th>Group update period $r_u$</th>
<th>Variance of participation</th>
<th>Variance of local accuracy</th>
<th>Accuracy of global test</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>60.00</td>
<td>24.81</td>
<td>94.93%</td>
</tr>
<tr>
<td>10</td>
<td>58.40</td>
<td>24.93</td>
<td>95.08%</td>
</tr>
<tr>
<td>20</td>
<td>53.94</td>
<td>19.71</td>
<td>94.51%</td>
</tr>
<tr>
<td>50</td>
<td><strong>43.42</strong></td>
<td><strong>16.36</strong></td>
<td><strong>95.46%</strong></td>
</tr>
</tbody>
</table>
# Analysis of Hyperparameter Selection for FedGLCE

## Table V

<table>
<thead>
<tr>
<th>Group update period $r_u$</th>
<th>CIFAR-10-Fed($\alpha = 10$)</th>
<th>CIFAR-10-Fed($\alpha = 0.1$)</th>
<th>CIFAR-10-Fed($\alpha = 0.01$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Var (par)</td>
<td>Var (local)</td>
<td>Acc (global)</td>
</tr>
<tr>
<td>5</td>
<td>249.92</td>
<td>35.80</td>
<td>71.40%</td>
</tr>
<tr>
<td>10</td>
<td>262.56</td>
<td>38.94</td>
<td>70.79%</td>
</tr>
<tr>
<td>20</td>
<td>265.04</td>
<td>38.79</td>
<td>69.84%</td>
</tr>
<tr>
<td>50</td>
<td>270.86</td>
<td>44.25</td>
<td>69.83%</td>
</tr>
</tbody>
</table>

## Table VI

<table>
<thead>
<tr>
<th>Group update period $r_u$</th>
<th>CIFAR-10-Fed($\alpha = 10$)</th>
<th>CIFAR-10-Fed($\alpha = 0.1$)</th>
<th>CIFAR-10-Fed($\alpha = 0.01$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Var (par)</td>
<td>Var (local)</td>
<td>Acc (global)</td>
</tr>
<tr>
<td>5</td>
<td>1096.90</td>
<td>29.94</td>
<td>71.12%</td>
</tr>
<tr>
<td>10</td>
<td>1126.62</td>
<td>36.00</td>
<td>70.54%</td>
</tr>
<tr>
<td>20</td>
<td>1134.32</td>
<td>35.29</td>
<td>71.02%</td>
</tr>
<tr>
<td>50</td>
<td>1165.22</td>
<td>41.05</td>
<td>70.94%</td>
</tr>
</tbody>
</table>
## Experiment

### Analysis of Client Participation

<table>
<thead>
<tr>
<th>Comparison approaches</th>
<th>Evaluation metrics</th>
<th>Dataset</th>
<th>MNIST-Fed</th>
<th>CIFAR-10-Fed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dir(10)</td>
<td>Dir(0.1)</td>
</tr>
<tr>
<td>FedAvg</td>
<td>64.30</td>
<td>1323.4</td>
<td>1326.86</td>
<td>1322.5</td>
</tr>
<tr>
<td>FedDrop</td>
<td>210.02</td>
<td>1887.22</td>
<td>1900.14</td>
<td>1818.46</td>
</tr>
<tr>
<td>TiFL</td>
<td>$Var(Fre)$</td>
<td>17.02</td>
<td>1537.14</td>
<td>1527.5</td>
</tr>
<tr>
<td>FedSS</td>
<td>78.36</td>
<td>1258.06</td>
<td>1259.86</td>
<td>1252.56</td>
</tr>
<tr>
<td>FedGLCE</td>
<td>53.94</td>
<td>1134.52</td>
<td>1022.92</td>
<td>1094.64</td>
</tr>
</tbody>
</table>
## Analysis of Local Test Accuracy

<table>
<thead>
<tr>
<th>Comparison approaches</th>
<th>Evaluation metrics</th>
<th>MNIST-Fed</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CIFAR-10-Fed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dir(10)</td>
</tr>
<tr>
<td>FedAvg</td>
<td></td>
<td>24.02</td>
<td>34.29</td>
</tr>
<tr>
<td>FedDrop</td>
<td>Var(Acc)</td>
<td>23.33</td>
<td>37.74</td>
</tr>
<tr>
<td>TiFL</td>
<td></td>
<td>15.77</td>
<td>38.79</td>
</tr>
<tr>
<td>FedSS</td>
<td></td>
<td>20.40</td>
<td>37.56</td>
</tr>
<tr>
<td><strong>FedGLCE</strong></td>
<td></td>
<td>19.71</td>
<td>35.29</td>
</tr>
</tbody>
</table>
Experiment

Total Hours of Federated Training

Experimental results of total federated learning training hours for different client grouping approaches.
FedGLCE can **balance the participation of clients** in the local federated training through the dynamic client grouping approach based on local computing efficiency under the premise of **ensuring the local test accuracy**.

**Performance biases** of the global model on clients are improved.

FedGLCE reduces the **total training time** of clients in federated learning.
Thanks for your attention!