POSTER: TOWARDS EFFECTIVE CLUSTERED FEDERATED LEARNING: A PEER-TO-PEER FRAMEWORK WITH ADAPTIVE NEIGHBOR MATCHING

Zexi Li [1] **Chao Wu** [2]

ABSTRACT

In federated learning (FL), clients may have diverse objectives, and merging all clients' knowledge into one global model will cause negative transfers to local performance. Thus, clustered FL is proposed to group similar clients into clusters and maintain several global models. Nevertheless, current clustered FL algorithms require the assumption of the number of clusters, and they are not effective enough to explore the latent relationships among clients. However, we take advantage of peer-to-peer (P2P) FL, where clients communicate with neighbors without a central server and propose an algorithm that enables clients to form an effective communication topology decentralized without assuming the number of clusters.

1 METHODS

1.1 Metrics for Measuring Client Similarity

Metrics for measuring the consistency of optimization objectives are needed to enable clients to select same-cluster peers and filter out outliers. Loss evaluation is a simple metric, commonly used in the literature (Ghosh & et al; Onoszko & et al). By evaluating other clients' models on local data, the model with a lower loss value is more likely to have a similar learning task. For P2P FL, we can formulate the similarities based on loss as

$$\mathrm{s}_{i,j} = 1/F_i\left(\mathbf{w}_j^t, \xi_i\right), \tag{1}$$

where $\mathrm{s}_{i,j}$ is similarity between client i and j, $\xi_i \sim \mathcal{D}_i$, $\mathcal{D}_i$ refers to the local data of client i. Since this metric is simple, we adopt this metric in our PANM (named as PANMLoss).

Additionally, we develop a more efficient metric based on gradients and accumulated weight updates. As used in centralized clustered FL (Sattler & et al), the cosine similarity of gradients is adopted in similarity measurement, as

$$\cos\theta_{i,j}^1 = \frac{\langle \mathbf{w}_i^t - \mathbf{w}_i^{t-1}, \mathbf{w}_j^t - \mathbf{w}_j^{t-1} \rangle}{\parallel \mathbf{w}_i^t - \mathbf{w}_i^{t-1} \parallel \cdot \parallel \mathbf{w}_j^t - \mathbf{w}_j^{t-1} \parallel}, \tag{2}$$

where θ^1 refers to the angle of vectorized gradients, $\mathbf{w}_i^t - \mathbf{w}_i^{t-1}$ refers to the gradients in the local updates of round t. But in P2P FL, without the central server, client model weights diverge since the first round, so the measurement of gradients will be noisy. We notice the accumulated weight updates from the initial model can signify the history optimization directions, and the cosine similarity of the weight updates can imply the consistency of objectives, as

$$\cos\theta_{i,j}^2 = \frac{\langle \mathbf{w}_i^t - \mathbf{w}_0, \mathbf{w}_j^t - \mathbf{w}_0 \rangle}{\parallel \mathbf{w}_i^t - \mathbf{w}_0 \parallel \cdot \parallel \mathbf{w}_j^t - \mathbf{w}_0 \parallel}. \tag{3}$$

According to Equation (2) and (3), we combine the two terms to formulate our new metric as

$$\mathrm{s}_{i,j} = \alpha \cos\theta_{i,j}^1 + (1-\alpha)\cos\theta_{i,j}^2, \tag{4}$$

where α is the hyperparameter controlling the weight of two cosine functions. Notably, our new metric is robust and effective in the P2P FL setting, and we refer it as PANMGrad.

1.2 Confident Neighbor Initialization

Based on the similarity metrics mentioned in the last subsection, we can devise our P2P FL algorithm **P**ersonalized **A**daptive **N**eighbour **M**atching (PANM).

In the first stage of P2P FL training, clients have to initialize their collaborative neighbors from randomly sampled peers ($\mathrm{C}_i^t, |\mathrm{C}_i^t| = l$). We propose a confident method: the Confident Neighbor Initialization (CNI) algorithm. In CNI, after the first round, we add the neighbors in the previous round (N_i^{t-1}) to the candidate list (N) in the current round (as shown in Equation 5), consequently, the confidence of same-cluster neighbors increases over round.

$$\begin{aligned} \mathrm{N}_i^t &= \arg\max_{\mathrm{N}} \sum_{j \in \mathrm{N}} \mathrm{s}_{i,j} \\ s.t.\ \mathrm{N} &\subsetneqq \mathrm{C}_i^t \cup \mathrm{N}_i^{t-1}, t > 1, |\mathrm{N}| = k. \end{aligned} \tag{5}$$

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China [2]School of Public Affairs, Zhejiang University, Hangzhou, China. Correspondence to: Chao Wu <chao.wu@zju.edu.cn>.

Proceedings of the 5^{th} MLSys Conference, Santa Clara, CA, USA, 2022.

1.3 Heuristic Neighbor Matching

After confident neighbor initialization, we enable clients to have few neighbors with high precision of being same-cluster. For clustered FL, the recall of clustering is also essential since each client needs to find the whole community with the same objective. Therefore, in stage two of PANM, we propose a more effective method based on expectation maximization (EM).

It is obvious that for a client, the same-cluster clients may have high similarities while the ones of the different-cluster are low, so we assume the similarities obey two distinct Gaussian distributions, thus we can formulate it into a Gaussian Mixture Model (GMM) problem. We devise our Heuristic Neighbor Matching (HNM) algorithm based on EM. In each round, client i randomly samples neighbor candidate list $\mathrm{C}_i^t(|\mathrm{C}_i^t| = l)$ from non-neighbor clients and also samples a selected neighbor list $\mathrm{S}_i^t(|\mathrm{S}_i^t| = l$ if $|\mathrm{N}_i^t| > l$, else $\mathrm{S}_i^t = \mathrm{N}_i^t)$ from neighbor clients N_i^t; client i communicate with these clients and compute similarities $y_j = \mathrm{s}_{i,j},\ j \in \mathrm{M}_i^t = \mathrm{C}_i^t \cup \mathrm{S}_i^t$. There are two Gaussian distributions in these similarities, the one with higher mean center refers to the same-cluster clients ($\mathcal{N}(\mu_0, \sigma_0^2)$), another one refers to the different-cluster ($\mathcal{N}(\mu_1, \sigma_1^2)$). Assuming the observed data $y_j,\ j \in \mathrm{M}_i^t$ are generated by GMM:

$$\Pr(y|\Theta) = \sum_{r=0}^{1} \beta_r \phi(y|\Theta_r). \tag{6}$$

Here, β_r refers to the probability that y is generated by distribution r, and $\Theta = (\beta_0, \beta_1; \Theta_0, \Theta_1)$. Our target is using EM algorithm to estimate the distribution identities of y_j, given by

$$\gamma_{j,r} = \begin{cases} 1, \text{ if } j \text{ belongs to distribution } \mathcal{N}_r \\ 0, \text{ otherwise.} \end{cases}$$

where $j \in \mathrm{M}_i^t,\ r \in \{0, 1\}$. Knowing that the EM algorithm is sensitive to initialization, with the prior knowledge that most of the clients in S_i^t are same-cluster (it is also possible that it includes outliers), so we can initialize a better parameter as

$$\gamma_{j,r}^{(1)} = \begin{cases} 1, j \in \mathrm{S}_i^t \text{ and } r = 1, \text{ or } j \in \mathrm{C}_i^t \text{ and } r = 0 \\ 0, \text{ otherwise.} \end{cases}$$

While the latent variable is the distribution parameters: $\Theta_0 = (\mu_0, \sigma_0), \Theta_1 = (\mu_1, \sigma_1)$, so the complete data is

$$(y_j, \Theta_0, \Theta_1),\ j \in \mathrm{M}_i^t.$$

Then we formulate the expectation function Q, based on the log likelihood function of complete data,

$$\begin{aligned} Q(\gamma, \gamma^{(a)}) &= \mathbb{E}[\log \Pr(y, \Theta|\gamma)|y, \gamma^{(a)}] \\ &= \sum_{r=0}^{1} \Big\{ n_r \log \mathbb{E}\beta_r + \sum_{j \in \mathrm{M}_i^t} \gamma_{j,r} [\log(\frac{1}{\sqrt{2\pi}}) - \log \mathbb{E}\sigma_r \\ &\quad - \frac{1}{2\mathbb{E}\sigma_r^2}(y_j - \mathbb{E}\mu_r)^2] \Big\}, \end{aligned} \tag{7}$$

where $n_r = \sum_{j \in \mathrm{M}_i^t} \gamma_{j,r}$.

E-step: Now we need to estimate $\mathbb{E}(\mu_r, \sigma_r, \beta_r)$, notated as $\hat{\mu}_r,\ \hat{\sigma}_r,\ \hat{\beta}_r$.

$$\hat{\mu}_r = \frac{\sum_{j \in \mathrm{M}_i^t} \gamma_{j,r} y_j}{n_r}, \hat{\beta}_r = \frac{n_r}{|\mathrm{M}_i^t|}, \hat{\sigma}_r^2 = \frac{\sum_{j \in \mathrm{M}_i^t} \gamma_{j,r}(y_j - \hat{\mu}_r)^2}{n_r},$$

where $r \in \{0, 1\}$.

M-step: Iterative M-step is to find the maximum of the function $Q(\gamma, \gamma^{(a)})$ with respect to $\gamma^{(a)}$, as to set $\gamma^{(a+1)}$ in the next iterative epoch

$$\gamma^{(a+1)} = \arg\max_{\gamma} Q(\gamma, \gamma^{(a)}). \tag{8}$$

We use the following function to maximize expectation, since y_j more likely belongs to $\mathcal{N}_0$ if $\beta_0 \phi(y_j|\Theta_0) > \beta_1 \phi(y_j|\Theta_1)$, vice versa

$$\begin{aligned} \gamma_{j,r}^{(a+1)} &= \mathbb{1}\left\{ r = \arg\max_r \frac{\hat{\beta}_r \phi(y_j|\hat{\Theta}_r)}{\sum_{c=0}^{1} \hat{\beta}_c \phi(y_j|\hat{\Theta}_c)} \right\} \\ &\quad j \in \mathrm{M}_i^t, r \in \{0, 1\}. \end{aligned} \tag{9}$$

Repeat the E-step and M-step until $\gamma^{(a+1)} = \gamma^{(a)}$. Then we obtain the estimated same-cluster neighbors in this round notated as H_i^t, where $\gamma_{j,0} = 1,\ j \in \mathrm{H}_i^t,\ \mathrm{H}_i^t \subseteqq \mathrm{M}_i^t$, then we update our neighbor list

$$\mathrm{N}_i^{t+1} = (\mathrm{N}_i^t - \mathrm{S}_i^t) \cup \mathrm{H}_i^t. \tag{10}$$

By HNM algorithm, clients can continually update their neighbors, adding new same-cluster clients and removing outliers in neighbor list.

REFERENCES

Ghosh, A. and et al. An efficient framework for clustered federated learning. In *NeurIPS 2020*.

Onoszko, N. and et al. Decentralized federated learning of deep neural networks on non-iid data. *arxiv*.

Sattler, F. and et al. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *TNNLS*.