Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9045

Client-Lightweight Soft Clustering Federated Learning

XinJie Liu. DingKai Hu*
Qingdao University, Qingdao 266071, China
Email: liuxinjie1@qdu.edu.cn

How to cite this paper: Liu, X. J., & Hu, D. K. (2026). Client-Lightweight Soft Clustering Federated Learning. Journal of Computer Science and Frontier Technologies, 2(3), 19-26. ISSN Print: 3104-4204; ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9045
Published: 2026-02-28

Abstract

Clustered federated learning is a highly effective approach to addressing the non-independent and identically distributed (Non-IID) problem in conventional federated learning. Soft clustering federated learning allows clients to belong to multiple clusters, which not only captures the common characteristics of clients within the same cluster but also fully adapts to the unique properties of each client's local data. This significantly improves model personalization and generalization ability, and greatly enhances model performance and training efficiency in Non-IID scenarios. Recent advanced soft clustering federated learning methods no longer rely on client model parameters for clustering. Instead, they generate asymmetric similarities via privacy-preserving set intersection operations on clients' local datasets, which are then sent to the server for clustering. This avoids the privacy risks associated with uploading model parameters from clients. In this work, we optimize this clustering paradigm by adopting locality-sensitive hashing and an improved privacy-preserving set intersection scheme. Our method substantially reduces the communication and computational overhead among clients while maintaining competitive model performance.

Keywords

Federated Learning; Soft Clustering; Private Set Intersection

1. Introduction

Unlike traditional hard clustering federated learning [1], soft clustering federated learning allows clients to belong to multiple clusters. By fusing the model parameters of multiple clusters weighted by membership degrees, it can not only capture the common characteristics of clients within the same cluster but also fully adapt to the unique properties of each client's local data, thereby significantly improving model personalization and generalization ability. By adapting to the mixed distribution characteristics of local data, it greatly enhances model personalization and training performance in Non-IID scenarios.

However, privacy leakage remains an inherent issue in federated learning, and soft clustering federated learning must also address privacy concerns. Moreover, soft clustering federated learning still requires model parameters for clustering and may additionally need to compute membership degrees. The computation of membership degrees often relies on local data distribution or model parameter similarity, which exposes more private information compared with conventional soft clustering federated learning.

Therefore, Xu et al. [2] proposed a privacy-enhanced and robust soft clustering federated learning framework (ProCFL). This scheme abandons clustering based on similarity calculated from model parameters. Instead, it uses a novel method to compute similarity among clients via privacy-preserving set intersection over their local datasets. The resulting similarity is asymmetric, preventing the server from inferring inter-client similarities.

To achieve soft clustering, they replace direct membership computation by reformulating the clustering problem as a weighted set cover problem. This approach selects the most valuable clusters for training and aggregation, while allowing clients to belong to multiple clusters. Nevertheless, computing similarity based on the intersection of local data samples has limitations: it cannot adequately reflect the distribution of local data. Furthermore, the adopted RSA-based private set intersection (RSA-PSI) [3] scheme incurs high communication overhead among clients. Therefore, in response to the above issues, our specific contributions are as follows:

- We use locality-sensitive hashing to compress the local data distribution vectors of clients, and perform privacy-preserving set intersection on the compressed binary vectors (0-1 vectors). This replaces the original privacy-preserving set intersection based on raw local data samples, reducing the communication overhead among clients;
- We adopt the more advanced private set intersection scheme, VOLE-PSI, which is more suitable for large-scale multi-feature datasets;
- Experiments demonstrate that our scheme achieves the same accuracy as ProCFL with lower overhead.

2. Related Work

Since the Google team proposed federated learning [4], a prominent challenge has been the non-independent and identically distributed (Non-IID) problem [4], [5], which is unavoidable for the practical deployment of federated learning. Therefore, an increasing number of researchers have sought to address this issue.

Clustered federated learning is one approach to addressing this problem. Ghosh et al. [10] propose a general algorithmic framework that addresses data heterogeneity via clustering and solves the Byzantine machine problem in heterogeneous federated learning. However, this algorithm also adopts K-means clustering, which requires

the number of clusters to be pre-specified. He et al. [11] present a hierarchical federated model embedding method, where they cluster participants using the outputs of local models on a global shared dataset. Sttler et al. [1] proposed the formal concept of federated clustered learning, which adopts iterative clustering. In each iteration, nodes are grouped into different clusters according to the similarity of their parameters. This scheme does not require predefining the number of clusters. Ye et al. [12] further apply clustering techniques to federated learning tasks in the recommendation domain and propose a framework called CPF-POI. In this framework, clustering is used to group participants with similar features, thereby improving the performance of personalized recommendation and optimizing the efficiency of knowledge sharing.

Marfoq et al. [6] proposed a framework similar to soft clustering federated learning. Assuming that client data follows a mixture distribution, they designed a federated expectation-maximization algorithm using an expectation-maximization framework to estimate mixture distribution coefficients based on training loss. However, this algorithm requires clients to perform local updates for all clusters in each round, resulting in high training overhead. It also assumes that the loss function takes a special form and that the marginal distributions of all components remain consistent, which is inconsistent with real-world scenarios. Ruan et al. [7] proposed FedSoft, a soft clustering federated learning method that allows each local dataset to follow a mixture of multiple source distributions while training local personalized models and high-quality cluster models. By using proximal updates to limit client workloads and requiring only one optimization task from a subset of clients in each communication round, it is more efficient than the method by Marfoq et al. Following the ideas of Marfoq and Ruan et al., Lin et al. [8] proposed FedSPD, a decentralized soft clustering federated learning approach. Its advantage lies in maintaining an independent model for each cluster and allowing clients to update only one cluster model per round using data associated with the corresponding cluster, unlike FedSoft by Ruan et al., which adopts a proximal objective function and uses all client data for model updates in each training round. Nevertheless, the decentralized nature leads to frequent communication among clients. Additionally, Li et al. innovatively combined the advantages of soft clustering and IFCA and proposed the FLSC algorithm to solve federated learning problems under heterogeneous training data. The algorithm designs a novel client local training scheme and a central server model fusion strategy based on overlapping client clustering. However, most of these soft clustering federated learning methods suffer from the drawback of requiring prior knowledge such as a predefined number of clusters.

3. Methodology

Our scheme is shown in Algorithm 1. Relevant similarity calculations are completed

in the initial round, followed by clustering. At $t = 0$, all clients first generate multiple pseudo-random vectors of uniform length K using d shared seeds, and construct them into a pseudo-random matrix $W \in R^{d*k}$ that follows the distribution $N(0,1)$. Next, each client multiplies its local data distribution ratio vector by this pseudo-random matrix to obtain the compressed vector S_i. Due to the properties of locality-sensitive hashing, this vector can retain certain characteristics of the original vector to a certain extent, such as the cosine similarity between vectors. Then, to obtain the similarity among clients through privacy-preserving set intersection between clients, each client binarizes the generated S_i: for each element in the vector, set it to 1 if positive, and to 0 if negative or zero. After that, the client records the indices of positions with value 1 in the binarized S_i, forming a set $S_i = j \in [0, k] | h_{ij} = 1$. Each client then performs privacy-preserving set intersection with other clients via VOLE-PSI [9], obtaining the count of positions where both their S_{other} vectors have value 1. Each client further divides this count by the number of 1s in its own local S_i vector, which serves as the asymmetric similarity ISM (The term ISM is named after [2] , although they derived it using local dataset samples, whereas we adopt the compressed local data distribution vectors of clients.) value with respect to that client. Each client thus obtains an ISM similarity vector and sends it to the server. The server aggregates all ISM similarity vectors to form the ISM matrix, and then employs the WSCP [2] algorithm to compute and obtain the final clustering result.

For $t >= 0$, the server performs an aggregation operation in each round. Clients first conduct local training based on the initial model parameters or the model parameters sent by the server in the previous round, and add differential privacy noise to the updated model parameters to prevent the server from inferring client privacy. The server then aggregates the model parameters uploaded by clients; it can perform global aggregation in the first few rounds and carry out corresponding cluster-based aggregation in later rounds.

Algorithm 1. The overall algorithm of our scheme.

Algorithm: Client-Lightweight Clustered Federated Learning Algorithm
Input: N: The total number of clients， $P_i = [p_1^i, p_2^i, p_3^i, \dots, p_d^i]$: The local data distribution vector of client ($i \in [1, N]$， where d denotes the total number of data labels per client), T: The total training rounds, k: the desired length of the hash code, G^0: initial model parameters, $U = \{c_1, \dots, c_N\}$: The initial client set, where each subset is the client c_i itself.
Output：G_{c_i}： model parameters of each cluster
1:For $t \in [0, T-1]$ do:
2: If $t = 1$ do:
3: For i ∈ [1, N] do:
4: Client c_i do:
5: generate a unified random matrix $W \in R^{d*k}, W \sim N(0,1)$;
6: compute $S_i = P_i * W$;
7: Generate a vector H_i of length $\lvert S_i \rvert$ initialized to 0 $(H_i = [0,0, \dots, 0])$;
8: For $j \in [1, \lvert S_i \rvert]$ do:
9: If $s_{ij} \geq 0$:

10: $h_{ij} = 1$;
11: Else:
12: $h_{ij} = 0$;
13: $S_i = \{j \in [0,k] \mid h_{ij} = 1\}$;
14: Each client performs VOLE-PSI(S_i, S_{other}) with every other client;
15: Obtain the similarity vector ISM and send it to the server;
16: Server do:
17: Obtains the new clustering result U= WSCP (ISM matrix, U), where $U = \{c_1, .., c_{\|U\|}\}$;
18: For i ∈ [1, N] do:
19: Client c_i do:
20: $g_i \leftarrow localtrain(G^t \text{ or } G_{c_i})$;
21: Send g_i to the server via differential privacy;
22: Server do:
23: Aggregate the model parameters G^t or G_{c_i} for each cluster $c_i (i \in [1, \|U\|])$.
24: Send G^t or G_{c_i} to the corresponding clients.
25:Return $G_{c_i} (i \in [1, \|U\|])$

4. Experiments

To evaluate the performance of the proposed scheme, this paper conducts comparisons from two dimensions: overhead and model accuracy. The datasets employed include CIFAR-10 (60k samples, 10 labels), FMNIST (70k samples, 10 labels), EMNIST (814,255 samples, 62 labels), ImageNet-1K (1.33M samples, 1k labels), and ImageNet-21K (14M samples, 21,841 labels), which are typical image classification datasets used in PSI performance evaluation. We conducted Model Evaluation on datasets such as CIFAR-10 and Fashion-MNIST (FMNIST): for CIFAR-10, we used a convolutional neural network with the structure Conv2d(3→6→5)→ReLU→MaxPool2d(2×2)→Conv2d(6→16→3)→ReLU→MaxPool2d(2×2)→ReLU→Linear(400→120)→ReLU→Linear(120→84)→ReLU→Linear(84→10), while for Fashion-MNIST, we used a CNN structured as Conv2d (1→16→3)→ReLU → MaxPool2d(2 × 2) → Conv2d(16 → 32 → 3) → ReLU → MaxPool2d(2 × 2) → Linear(1568 → 128) → ReLU → Linear(128 → 10); all clients employed the SGD optimizer with 1 local epoch, a learning rate of 0.01, and a batch size of 256. Assume that 100 clients participate in dataset partitioning, and each client holds the same number of local data samples, which are non-independent and identically distributed (Non-IID). We adopt the Dirichlet distribution to construct the Non-IID setting with the parameter b=0.5。The key length of RSA-PSI is set to 2048 bits. The statistical security parameter of VOLE-PSI is set to 40. We use Gaussian noise for differential privacy, with σ set to 0.006.

4.1. Overhead Comparison

According to [9], different privacy-preserving set intersection protocols adopt different implementation methods, and their running times vary greatly. Therefore, comparing communication traffic is a reasonable way to evaluate the overhead of different schemes. In this section, we compare the communication overhead of the

VOLE-PSI used in our scheme with the RSA-PSI adopted in the ProCFL scheme. The comparison is performed on datasets of different scales: FMNIST, EMNIST, ImageNet-1K, and ImageNet-21K. Since the ProCFL scheme uses the clients' local datasets while our scheme uses the clients' local data distributions, we set up the ProCFL scheme by assuming 100 clients and evenly dividing the dataset among them. For our scheme, we directly conduct privacy-preserving set intersection based on the length of the data distribution vectors.

For ProCFL, the sizes of the sets used for intersection computation by each client on the FMNIST, EMNIST, ImageNet-1K, and ImageNet-21K datasets are 700, 8000, 12800, and 140000, respectively. For our scheme, the sizes of the sets used for intersection computation by each client on the aforementioned datasets are 10, 62, 1000, and 21841, respectively. The two clients performing the set intersection act as the client and the server, respectively. The experimental results are shown in Table 1. Our scheme, which combines locality-sensitive hashing on data distributions with VOLE-PSI overhead, outperforms RSA-PSI used in ProCFL on all datasets, and our approach has greater advantages on large-scale datasets. This is because our scheme computes the intersection over data distributions, whereas ProCFL performs set intersection based on the raw local data of clients.

Table 1. Communication overhead comparison between our scheme and ProCFL.

Dataset	Scheme	Entity		Total
		Client	Server	
FMNIST	RSA-PSI	0.34 MB	0.34 MB	0.68 MB
	OUR	0.0431 MB	0.4996 MB	0.5427 MB
EMNIST	RSA-PSI	3.9061 MB	3.9061 MB	7.8122 MB
	OUR	0.0434 MB	0.5009 MB	0.5443 MB
ImageNet-1K	RSA-PSI	6.2499 MB	6.2499 MB	12.4998 MB
	OUR	0.0659 MB	0.5250 MB	0.5909 MB
ImageNet-21K	RSA-PSI	68.3578 MB	68.3578 MB	136.7156 MB
	OUR	0.2290 MB	0.8263	1.0553 MB

4.2. Model Evaluation

We evaluate our proposed scheme, ProCFL and FedAvg [4] on the CIFAR and FMNIST datasets. The validation approach adopted is to generate a dataset conforming to each client's local data distribution for each round to perform accuracy detection. The test set we use is the union of the training set and the test set. As shown in Figure 1, our proposed scheme outperforms both FedAvg and ProCFL. In Figure 2, the accuracy of our scheme and ProCFL is much higher than that of FedAvg, which demonstrates the advantage of clustering. Furthermore, on the CIFAR-10 dataset, our scheme achieves comparable accuracy with ProCFL.

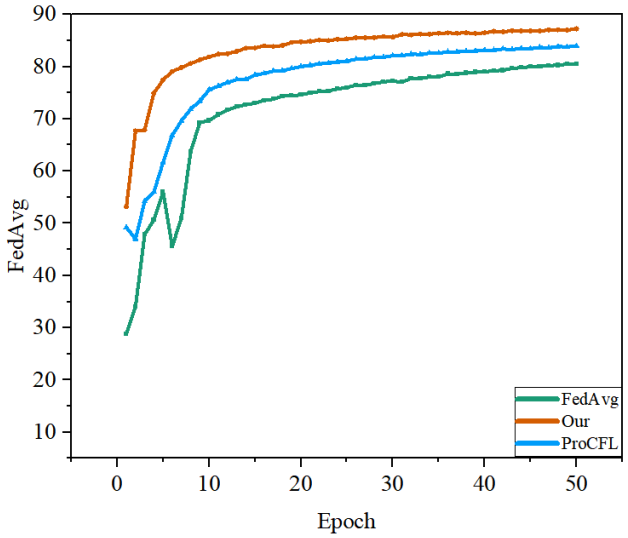

Figure 1. The accuracy of the schemes on the FMNIST dataset.

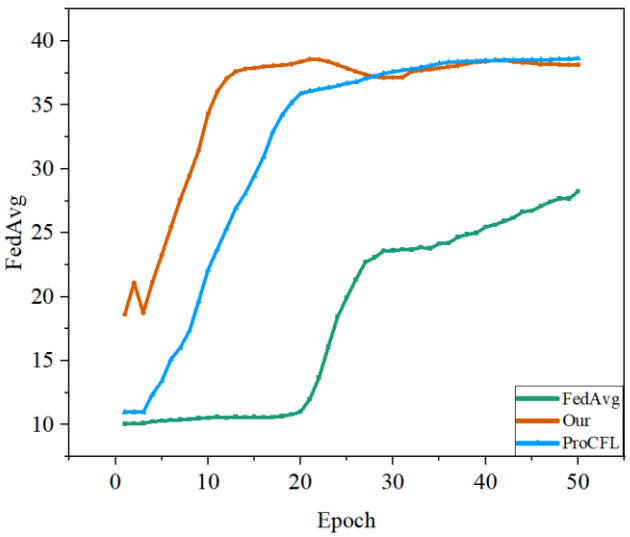

Figure 2. The accuracy of the schemes on the CIFAR-10 dataset.

5. Conclusion

In this paper, we propose a client-lightweight soft clustered federated learning framework. We use locality-sensitive hashing to compress the data distribution of each client, and then perform privacy-preserving set intersection among clients based on these compressed vectors. This replaces the method in ProCFL that conducts privacy-preserving set intersection using the local raw samples of clients. It significantly reduces the overhead for clients to compute asymmetric similarities with each other, especially on large-scale datasets. Furthermore, the final accuracy of our model is superior to or comparable with that of ProCFL and far exceeds FedAvg. In future work, we will continue to explore privacy preservation and

defense against malicious attacks for soft clustered federated learning, and strive to balance performance and overhead, so as to make soft clustered federated learning more practical and applicable to a wider range of scenarios.

References

[1] F. Sattler, K.-R. Müller, W. Samek, Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints, IEEE transactions on neural networks and learning systems 32 (8) (2020) 3710–3722.

[2] Y. Xu et al., "Towards Privacy-Enhanced and Robust Clustered Federated Learning," in IEEE Transactions on Mobile Computing, vol. 24, no. 8, pp. 7171-7188, Aug. 2025, doi: 10.1109/TMC.2025.3547149.

[3] Á. Kiss, J. Liu, T. Schneider, N. Asokan, and B. Pinkas, "Private set intersection for unequal set sizes with mobile applications," Privacy Enhancing Technol., vol. 2017, no. 4, pp. 177–197, 2017.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: International Conference on Artificial Intelligence and Statistics, 2016.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data (2018). doi:10.48550/ARXIV.1806.00582.

[6] Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, L.; and Vidal, R. 2021. Federated Multi-Task Learning under a Mixture of Distributions. International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021 (FL-ICML'21).

[7] Y. Ruan and C. Joe Wong, "FedSoft: Soft clustered federated learning with proximal local updating," in Proc. AAAI Conf. Artif. Intell.2022, pp. 8124-8131.

[8] I-Cheng Lin, Osman Yağan, and Carlee Joe-Wong. 2025. FedSPD: a soft-clustering approach for personalized decentralized federated learning. In Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence (UAI '25), Vol. 286. JMLR.org, Article 113, 2618–2641.

[9] Peter Rindal and Phillipp Schoppmann. 2021. VOLE-PSI: Fast OPRF and Circuit-PSI from Vector-OLE. In Advances in Cryptology – EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, 901–930. https://doi.org/10.1007/978-3-030-77886-6_31

[10] A. Ghosh, J. Hong, D. Yin, K. Ramchandran, Robust federated learning in a heterogeneous environment, arXiv preprint arXiv:1906.06629(2019).

[11] Y. He, D. Yan, F. Chen, Hierarchical federated learning with local model embedding, Engineering Applications of Artificial Intelligence 123 (2023) 106148.

[12] Z. Ye, X. Zhang, X. Chen, H. Xiong, D. Yu, Adaptive clustering based personalized federated learning framework for next poi recommendation with location noise, IEEE Transactions on Knowledge and Data Engineering 36 (5) (2024) 1843–1856. doi:10.1109/TKDE.2023.3312511.