2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS)

Federated Clustering with Unknown Number of Clusters

Rong Zou[1,a], Yunfan Zhang[2,b], Yiqun Zhang[1,2,c], Yang Lu[1,3,d], Mengke Li[1,4,e], Yiu-ming Cheung[1*]
[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China
[2]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[3]School of Informatics, Xiamen University, Xiamen, China
[4]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen, China
[a]rongzou@comp.hkbu.edu.hk, [b]3121008002@mail2.gdut.edu.cn, [c]yqzhang@gdut.edu.cn,
[d]luyang@xmu.edu.cn, [e]limengke@gml.ac.cn, [*]ymc@comp.hkbu.edu.hk

2024 6th International Conference on Data-driven Optimization of Complex Systems (DOCS) | 979-8-3503-7784-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/DOCS63458.2024.10704350

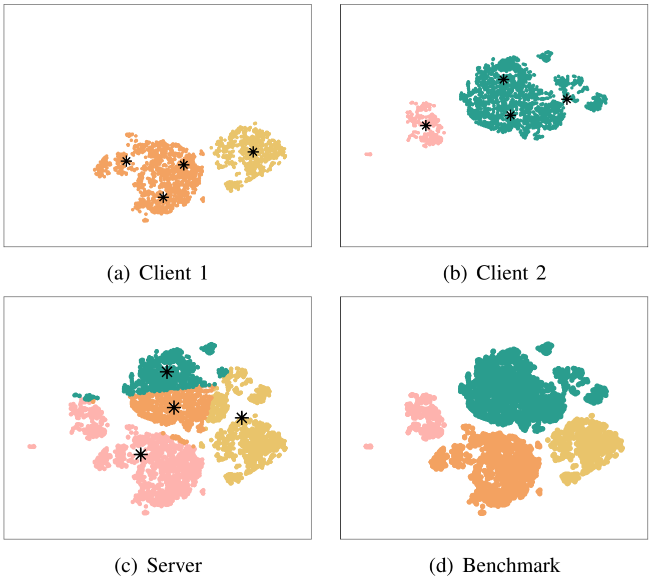
***Abstract*—Federated clustering is crucial to mining knowledge from unlabeled data distributed to multiple clients while preserving privacy. As there is no explicit learning supervision, clustering is considered a challenging federated learning task. Most existing works assume that the 'true' cluster number k^* is given to each client and server, which is far from a real federated learning scenario. Without the guidance of k^*, federated clustering becomes more challenging, rendering most existing solutions infeasible. We therefore propose a Federated Competitive and Cooperative Learning mechanism (FedCCL) to explore and fuse heterogeneous cluster distributions from clients automatically, and eventually form a global cluster partition, without requiring the cluster number to be given. We let the clients download seed points to explore their local distributions, which are then uploaded to the server for fusion. Different clients are allowed to compete on a single seed to form a consensus, while close seeds cooperate to represent a cluster. By iteratively homogenizing the cooperated seeds, a proper number of clusters will gradually emerge. Extensive experiments demonstrate the effectiveness of the proposed method.**

***Index Terms*—Federated Clustering, Competitive and Cooperative Learning, Unknown Number of Clusters**

I. Introduction

Federated learning aims to realize machine learning under constraints of privacy and security [1], [2], [3]. In unsupervised federated learning tasks, clustering that partitions a dataset into compact object clusters demonstrates great potential in mining data knowledge [4], [5]. However, the settings of federated learning bring great challenges to clustering, as labels are unavailable to explicitly guide the learning process. Most existing federated clustering attempts assume that the cluster number to seek is known in advance, and can be roughly categorized into one-shot [6] and iterative approaches based on the communication frequency between client and server [7].

One-shot federated clustering learns cluster distributions locally and passes the learned knowledge to the server for global cluster distribution aggregation. k-FED [8] adopts such a paradigm to explore more comprehensive global cluster distributions through one-shot aggregation of the non-Independent and Identically Distributed (non-IID) distributions learned by the clients. However, k-FED assumes that the proper numbers of clusters for each client are given, and different clients are with clearly separable cluster distributions that can be easily learned by the k-means-type algorithms [9], [10]. In general, since the aggregation is performed in only one shot, it cannot provide sufficient opportunity for the clients to interact and complement each other for more comprehensive cluster distribution exploration.


(a) Client 1

(b) Client 2

(c) Server

(d) Benchmark

Fig. 1. Direct use of existing iterative federated clustering methods on non-IID data. Even with a 'true' k^*, unexpected clustering results still easily occur.

Iterative federated clustering approaches facilitate sufficient interaction among the clients through the server by iteratively performing the following steps: (1) implement cluster distribution learning at the clients, (2) fuse privacy-protected distribution information at the server, and (3) send back the fused information to the clients for further tuning. To achieve a more comprehensive aggregation at the server, two independent works called F-FCM [11] and FFCM [12] sharing similar principles adopt fuzzy-c-means as the base clustering algorithm. The fuzzy object-cluster affiliation can

[*]Corresponding author

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.

more finely reflect the partition information of data objects, which can somewhat offset the information loss due to the privacy constraints of federated learning.

Nevertheless, all the above federated clustering approaches assume the cluster number to be given to all the clients and the server in advance, severely constraining their applicability. As shown in Fig. 1, direct use of the relatively powerful iterative FFCM by setting the cluster number of all the clients and server to the global k^* may easily cause unexpected clustering results due to the non-IID of clients.

To relieve the dependence on a given k^*, we propose a new federated clustering approach called Federated Competitive and Cooperative Learning (FedCCL). It can automatically explore the cluster distributions without knowing the number of clusters. To leverage the merits of Competitive and Cooperative Learning (CCL) [13], [14] to automatically select k^* in a federated setting, we propose an asynchronous cluster centroid interactive learning mechanism. It accumulates the update intensity of each seed point within different clients, and then passes to the server for competitive client-to-seed information fusion. To address the thorny case that a global cluster is composed of several sub-clusters from different clients, we let neighboring seed points share their update intensity to achieve a cooperative seed-to-seed information fusion. As a result, the representative seeds gradually absorb the surrounding ones until they are duplicated, i.e., trapped by the corresponding clusters. Extensive experimental evaluations demonstrate the effectiveness of FedCCL. The main contributions of this work are summarized into three points:

- We propose a new federated clustering approach that does not require the 'true' cluster number, and thus enhances the universality of current federated clustering.
- This paper attempts to address a realistic but challenging non-IID case, i.e., a global cluster is composed of non-overlapping sub-clusters from different clients.
- An asynchronous competitive cooperative learning mechanism is proposed, which facilitates sufficient interactive learning of non-IID clients under federated scenarios.

II. Related Work

Federated Clustering: As data privacy issues are rarely considered and the increasing harm caused by data security problems, federated clustering has gained increasing attention for its role in protecting personal data. Dennis [8] developed a one-shot federated clustering scheme, k-FED, which utilized heterogeneity between clients under the center separation hypothesis and thus weakened the cluster separation requirements for k-FED. However, it uses Floyd's k-means in each client under the assumption that k-FED knows the cluster number from every client in advance. Most recently, a density-based method HFDPC [15] introduces a similar density chain to mitigate the "domino effect" caused by multiple local peaks in a flow pattern dataset. However, its efficiency is non-ideal, as it further utilizes data dimension reduction and image encryption for partitioning. As far as we know, most existing federated clustering methods heavily rely on the 'true' cluster number k^*, which hinders their application.

Clustering with Unknown Cluster Number: For most clustering methods, the cluster number k is a crucial parameter. Different strategies have been proposed to implement clustering with unknown k. Many efforts have been made to develop approaches without using k, e.g., the rival penalization controlled competitive learning clustering (RPCCL) [16] approach and the competitive and cooperative learning approach [13]. Recently, a method called PCL-OC [17] has further extended the automatic cluster number selection to complex mixed data scenarios. They all require the frequent interaction among seed points representing clusters, to let the redundant seeds to be eliminated. However, the federated condition that restricts the direct interaction among clients brings great challenges to the automatic selection of k.

III. Proposed Method

We first define the research problem and then present the proposed algorithm as two parts: 1) ClientUA: Client-side Update Accumulation, and 2) ServerSI: Server-side Seeds Interaction. The overall pipeline is demonstrated in Fig. 2.

Assume a dataset $X = \{X^{[1]}, X^{[2]}, ..., X^{[g]}, ..., X^{[p]}\}$ is composed of data from p different clients, where client data $X^{[g]} = \{x_1^{[g]}, x_2^{[g]}, ..., x_{n^{[g]}}^{[g]}\}$ has $n^{[g]}$ samples and sample $x_i^{[g]}$ $= \{x_{i,1}^{[g]}, x_{i,2}^{[g]}, ..., x_{i,d}^{[g]}\}$ has d feature values, $i \in \{1, 2, ..., n^{[g]}\}$. Initially, each client employs a conventional clustering method, e.g. k-means++[18], to initialize k centroids, where k can be set as twice the value of k^* ('true' cluster number of the dataset X) or a relatively large value if the k^* is unattainable, thereby ensuring that the set k surpasses k^* and is therefore equal or greater than the 'true' cluster number for all clients. Subsequently, the centroids are transmitted to the server, and the $p * k$ centroids from p clients will be assigned to k global centroids $w_1, ..., w_k$ through the conventional clustering method, and then disseminated to the p clients.

ClientUA: To facilitate the fusion of non-IID information from different clients with unknown cluster numbers, update intensity $R^{[g]}$ is introduced to represent the sample distribution during client-side competitive learning. Additionally, we also need to compute the sample mean $b^{[g]}$, the average distance $z^{[g]}$, and the corresponding label $Q^{[g]}$, which are utilized as convergence information to support the calculation of convergence judgement function Z on the server.

For illustrative purposes, we take an arbitrary client g as an example. Assume client g has $n^{[g]}$ samples, k collections of sample label $Q_1, ..., Q_l, ..., Q_k$ and the corresponding k centroids $w_1, ..., w_l, ..., w_k$. However, since the data across each client is non-IID, the 'true' cluster number is less than or equal to the global set k, i.e., $k^{[g]*} \leq k$. For clarity, during the client-server iteration phase, $w_1, ..., w_k$ represent the global centroids received from the server, which remains unchanged throughout the operation process on the client side.

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.

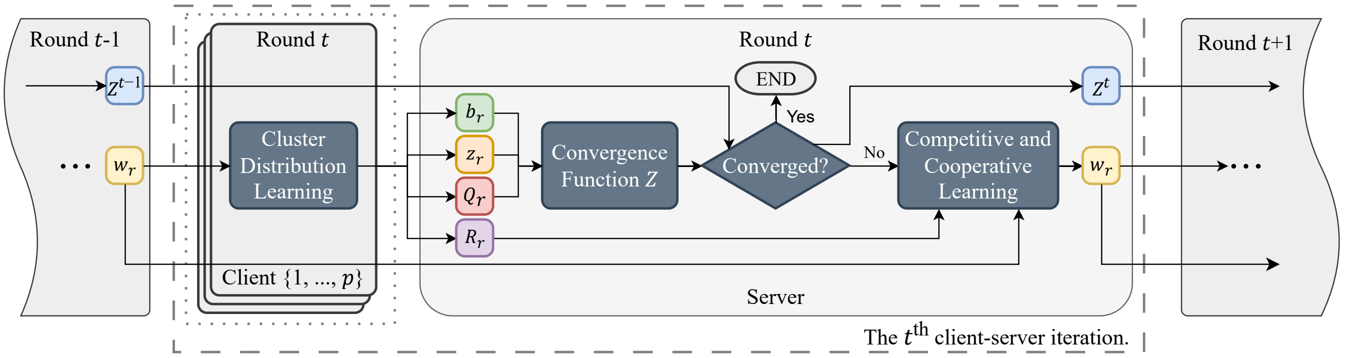


Fig. 2. Overview of the proposed FedCCL Algorithm.

The cluster to which the sample $x_i^{[g]}$ on client g belongs will be determined based on the indicator function

$$P(l \mid x_i^{[g]}) = \begin{cases} 1, & \text{if } l = arg\min_{g} \gamma_r \parallel x_i^{[g]} - w_r^{[g]} \parallel^2 \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $w_r^{[g]}$ is the r-th centroids in client g. That is, $x_i^{[g]}$ is classified into the l-th cluster ($Q_r^{[g]} = x_i^{[g]} \to Q_r^{[g]}$) if $P(l \mid x_i^{[g]}) = 1$, where $i \in \{1, 2, ..., n^{[g]}\}$. $\gamma_r^{[g]}$ is the associated win count:

$$\gamma_r^{[g]} = \frac{s_r^{[g]}}{\sum_{j=1}^{k^{[g]}} s_j^{[g]}}, \quad (2)$$

which represents the weight of $w_r^{[g]}$ under the sum of the win count of all centroids. $s_r^{[g]}$ denotes the cumulative win count of $w_r^{[g]}$ within a single iteration process. For the winner $w_c^{[g]}$, i.e. $P(c \mid x_i^{[g]}) = 1$, both win count $s_c^{[g]}$ and update intensity $R_{c,i}^{[g]}$ will be updated by

$$s_c^{[g]new} = s_c^{[g]old} + 1, \quad (3)$$

$$R_{c,i}^{[g]new} = R_{c,i}^{[g]old} + \eta(x_i^{[g]} - w_c^{[g]}), \quad (4)$$

while centroid $w_c^{[g]}$ remains unchanged. η is the learning rate.

Upon traversing all samples, we need to calculate convergence information. Nonetheless, given that the data across each client is non-IID within a federated environment, there exists a possibility that some clients may only have one single cluster. Specifically, the 'true' cluster number for client g, denoted as $k^{[g]*}$, equals 1. Under such circumstances, the computed result is meaningless as all samples within the client should be classified to the same unique centroid. Consequently, it is imperative to calculate the intermediate value of the convergence information for each client and transmit it to the server for the final convergence function. To this end, as one part of convergence information, the sample mean $b_r^{[g]}$ and the average distance $z_r^{[g]}$ from $b_r^{[g]}$ to the corresponding samples can be derived by

$$b_r^{[g]} = \sum_{i=1}^{|\mathrm{Q}_r|} x_i^{[g]} / |Q_r^{[g]}|, \quad z_r^{[g]} = \sum_{i=1}^{|\mathrm{Q}_r|} \parallel b_r^{[g]} - x_i^{[g]} \parallel^2, \quad (5)$$

where $x_i^{[g]} \in Q_r^{[g]}$.

ServerSI: The server primarily handles the aggregation of data received from clients, convergence function computation, and competitive and cooperative learning for global centroids, aiming to achieve a cooperative seed-to-seed information fusion. Upon receiving the sample mean b_r and the average distance z_r from each client, the server aggregates them by

$$b_r = \sum_{g=1}^{\mathrm{p}} \frac{|Q_r^{[g]}| b_r^{[g]}}{\sum_{g=1}^{\mathrm{p}} |Q_r^{[g]}|}, \quad z_r = \sum_{g=1}^{\mathrm{p}} \frac{|Q_r^{[g]}| z_r^{[g]}}{\sum_{g=1}^{\mathrm{p}} |Q_r^{[g]}|}. \quad (6)$$

Then the value of Z to judge the convergence is computed by

$$Z = \frac{1}{k} \sum_{i,j=1}^{\mathrm{k}} \min_{i \neq j} D_{i,j}, \quad D_{i,j} = \frac{z_i + z_j}{\sqrt{\parallel b_i - b_j \parallel^2}}, \quad (7)$$

where $D_{i,j}$ is the dissimilarity between cluster Q_i and Q_j. The convergence of the proposed FedCCL algorithm is judged according to the change of Z. When the change of Z falls below a predefined small threshold ε, i.e. $Z^t - Z^{t-1} < \varepsilon$, FedCCL is judged to be convergence. Should the results fail to converge, the server will perform Competitive and Cooperative Learning with the received results and update intensity. To specific, for winner w_r based on update intensity r_r, the collaborator C_r can be calculated by

$$C_r = C_r \cup \{w_j \mid \parallel w_r - w_j \parallel \leq \parallel w_r - (r_r/\eta + w_r) \parallel\}, \quad (8)$$

where $r_r \in R_r$. Collaborator C_r comprises all centroids, including w_r, that are in proximity to w_r. As delineated in Eq. (8), the distance requirement is contingent upon r_r. A larger r_r value implies that the centroid w_r has a broad sample distribution range across all clients, suggesting that the collaborator centroids need to be identified within a larger radius for better cooperation. The positions of all centroids from collaborator C_r will be moved accordingly based on server-side sample distribution which is computed from the update intensity r_i of w_r by

$$\begin{aligned} w_u &= w_u + \eta((r_i/\eta + w_r^{old}) - w_u) \\ &= w_u + r_i + \eta w_r^{old} - \eta w_u, \end{aligned} \quad (9)$$

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 FedCCL

```
Input: Initial cluster number $k$ with $k \gg k^*$, Dataset $X^{[1]}$, $X^{[2]}, ..., X^{[p]}$ for $p$ clients;
Output: The global centroids $w_1, ...w_{k_*}$.
for each client $g \in \{1, ..., p\}$ in parallel do
    Client $g$: Initialize $k$ centroids $w_1^{[g]}, ..., w_k^{[g]}$;
end for
Transfer initialization result to server;
Server: initialize $k$ global centroids $w_1, ..., w_k$;
Initialize $Z^t \leftarrow 0$;
repeat
    Transfer global centroids $w_1, ..., w_k$ to all $p$ clients;
    for each client $g \in \{1, ..., p\}$ in parallel do
        Client $g$: execute ClientUA;
    end for
    Transfer algorithm output to server;
    Update $Z^{t-1} \leftarrow Z^t$;
    Server: execute ServerSI;
until convergence
```

where $w_u \in C_r$ and $r_i \in R_r$. When all centroids and their collaborators are located in new positions, the server sends the centroids $w_1, ..., w_k$ back to each of the clients, and the algorithm returns to the ClientUA part to start a new iteration.

The design of Eq. (8) and (9) aims to enhance the adaptability of our proposed federated clustering method to various sample distributions. It also serves to prevent the occurrence of dead seeds. Considering the collaborators C_r, for w_r, Eq. (9) is transformed to $w_r = w_r + r_i + \eta w_r^{old} - \eta w_r$. That is, the movement of w_r is dominated by $r_i \in R_r$. For centroids other than w_r, the movement depends on both the update intensity and the position of w_r^{old}. In other words, all centroids in C_r will also tend to move closer to w_r when they transit to new positions according to the updated intensity. During the operation of traversing all centroids and their collaborators, neighboring centroids share their update intensity and achieve cooperative seed-to-seed information fusion. During the client-server iterative learning, the representative centroids will gradually absorb the surrounding less-representative ones until they are duplicated and trapped by the corresponding cluster distributions.

Overall FedCCL Algorithm: In FedCCL, the client-side mainly implements cluster distribution learning, and the server-side is responsible for privacy-protected distribution information fusion. The entire process is summarized in Algorithm 1. The time complexity of each iteration of FedCCL is $\mathcal{O}(kn^{[g]}dp + n^{[g]}k^2d)$, which is linear w.r.t. n.

IV. Experiments

Experimental Setup: Four experiments have been conducted to evaluate the proposed FedCCL: (1) Visualization, (2) Convergence Evaluation, (3) Clustering Performance Evaluation, and (4) Ablation Study.

Five counterparts have been compared. Federated Mean Shift (FMS) is a simple baseline federated clustering approach realized by replacing the fuzzy c-means [19] with the conventional mean shift clustering algorithm [20] for the FFCM [12]. DK++ [21] is a conventional distributed learning approach. Since it also conforms to the settings of federated learning, we also adopt it as a counterpart. Three state-of-the-art methods, i.e., the iterative learning approaches FFCM-avg1, FFCM-avg2 [12], and the one-shot learning approach k-FED [8], are also chosen for comparison. Hyper-parameters of the counterparts (if any) are set according to the corresponding source papers.

Thirteen datasets including two synthetic and eleven real benchmark datasets have been utilized. Statistics of the datasets are shown in Table I. All the public datasets are collected from the UCI machine learning repository [22]. All the datasets are pre-processed by omitting the objects with missing values. Two 2-D synthetic datasets are intuitively demonstrated in Fig. 3 using t-SNE [23], and objects belonging to the same true cluster are marked by the same color.

TABLE I
Statistics of experimental datasets.

No.	Dataset	Abbrev.	n	d	k^*
1	Synthetic Dataset 1	SD1	2300	2	4
2	Synthetic Dataset 2	SD2	2900	2	5
3	Drug Consumption	DC	1885	12	7
4	Avila	AL	10430	10	12
5	Abalone	AB	4178	8	29
6	Cancer	CC	570	31	2
7	Ecoli	EC	336	7	8
8	Seeds	SE	210	7	3
9	Parkinson	PA	195	22	2
10	Accent	AC	330	12	6
11	Sports Articles	SP	1000	59	2
12	Iris	IR	150	4	3
13	Segment	SG	2100	19	7

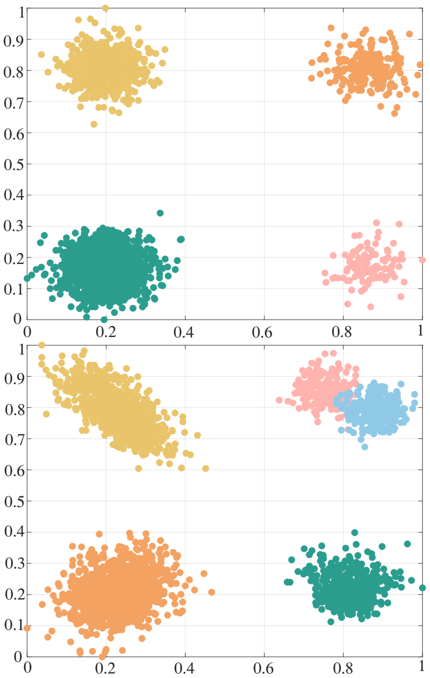
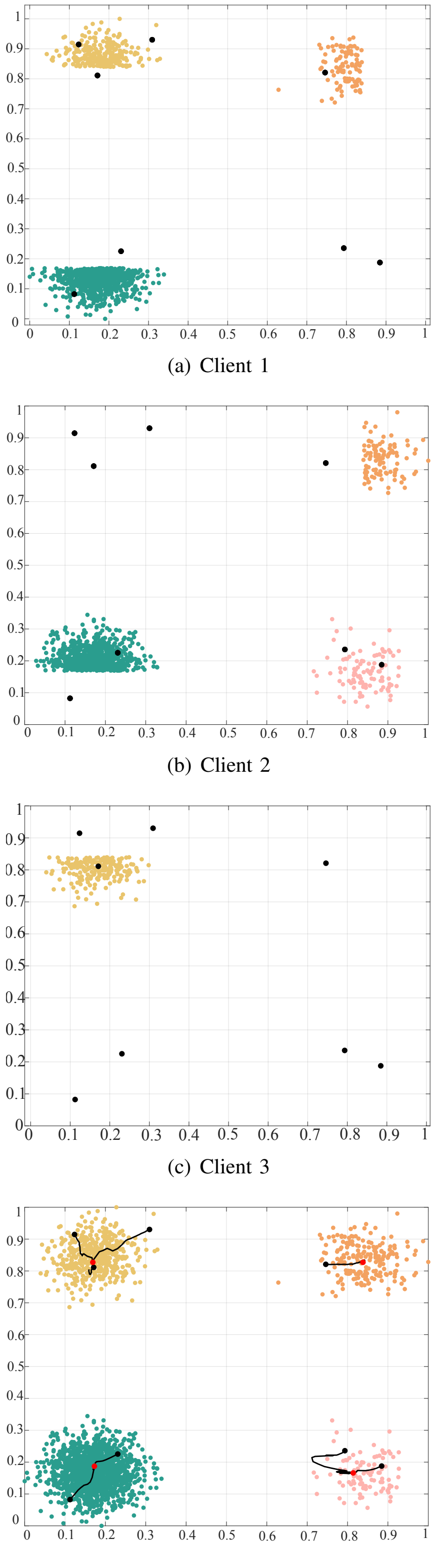


Fig. 3. Visualization of SD1 (upper) and SD2 (lower).

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.



(a) Client 1

(b) Client 2

(c) Client 3

(d) Clustering Process on Server

Fig. 4. Cluster centroids and the trajectories during the learning of FedCCL. Black and red dots indicate the initial positions and final positions of the centroids, respectively.

Two validity indices have been chosen. Silhouette Coefficient index (SC) [24] is a conventional and popular internal index, which simultaneously indicates the compactness of clusters and the dispersion among clusters, with its values in [-1,1]. Calinski-Harabasz index (CH) [25] computes the ratio of the average inter-centroid distance to the average object-centroid distance within clusters, with values ranging from (0,+∞). For both of them, a higher value indicates a better clustering performance.

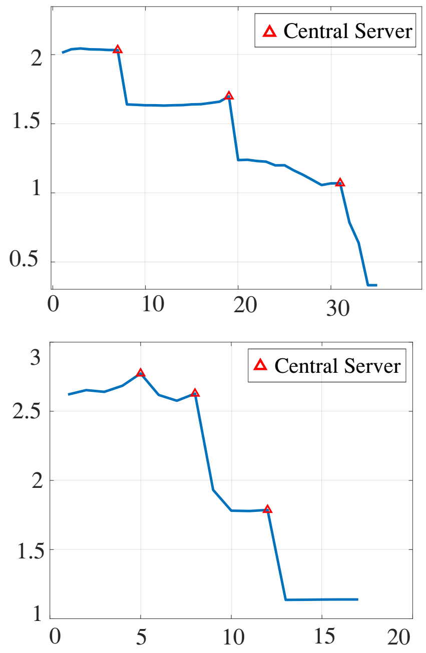


Fig. 5. Values of the FedCCL objective function on the SD1 dataset (upper) and DC dataset (lower). The red triangles mark the iterations of the server update.

TABLE II
CLUSTERING PERFORMANCE OF FEDCCL AND W/O SERVERSI.

Dataset	FedCCL	w/o ServerSI	Dataset	FedCCL	w/o ServerSI
SD1	19670.1	**20556.3**	SE	**254.5**	234.5
SD2	**19781.1**	18415.4	PA	**65.0**	64.6
DC	**831.8**	519.1	AC	**156.3**	89.3
AL	75.9	**1651.1**	SP	**455.3**	356.3
AB	**7910.1**	2997.4	IR	**330.4**	148.0
CC	**315.9**	289.0	SG	**1620.9**	949.3
EC	113.2	**116.4**			

Visualization: To intuitively validate the effectiveness of FedCCL, we split SD1 into three subsets for creating non-IID data for three clients as shown in Fig. 4(a)∼(c). Fig. 4(d) shows the distributions and trajectories of the centroids in the server. It can be observed that even though the three clients have completely non-overlapping distributions, FedCCL can still obtain the global cluster distributions through federated clustering. The trajectories also demonstrate that the asynchronous seed point updating mechanism of FedCCL effectively facilitates interaction among the seeds. After several iterations, redundant seeds are overlapped, indicating the same prominent global

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON OF SC PERFORMANCE ON 13 DATASETS.

Dataset	FedCCL	FFCM-avg1	FFCM-avg2	k-FED	DK++	FMS
SD1	**0.9718±0.0000**	0.5063±0.0241	0.5036±0.0215	0.8494±0.0000	0.5986±0.1798	0.8204±0.0000
SD2	**0.8766±0.0359**	0.4773±0.0261	0.4679±0.0324	0.7699±0.0000	0.6127±0.1046	0.7455±0.0423
DC	**0.5349±0.0000**	0.1675±0.1254	0.1476±0.1792	0.2104±0.0307	0.2884±0.0168	0.1812±0.0521
AL	**0.9750±0.0000**	0.2721±0.0921	0.2761±0.0701	0.0979±0.0266	0.2096±0.0194	0.4056±0.0142
AB	**0.5139±0.0228**	0.3664±0.3380	0.4863±0.2576	0.1853±0.0087	0.2314±0.0162	-0.2242±0.1359
CC	**0.5963±0.0664**	0.2873±0.0509	0.3051±0.0440	0.3809±0.0203	0.3778±0.0000	0.4624±0.0762
EC	**0.4739±0.0806**	0.3442±0.4041	0.4500±0.3016	0.3246±0.0073	0.2852±0.0370	0.3537±0.0468
SE	**0.5518±0.0399**	0.4321±0.0753	0.4321±0.0753	0.3754±0.0358	0.3229±0.0000	0.3605±0.0217
PA	**0.6284±0.1702**	0.4419±0.1549	0.4419±0.1549	0.4517±0.0328	0.2763±0.0000	0.6016±0.0392
AC	**0.6385±0.1369**	0.2656±0.1390	0.2358±0.1346	0.0992±0.0084	0.1831±0.0226	0.0777±0.0425
SP	0.5466±0.0036	0.2800±0.0516	0.2796±0.0507	0.5194±0.0400	0.3441±0.0028	**0.5844±0.0573**
IR	**0.6579±0.0181**	0.5672±0.0611	0.6119±0.2075	0.4818±0.0198	0.4955±0.0107	0.4487±0.0148
SG	**0.5702±0.0130**	0.3819±0.1113	0.3865±0.1020	0.3117±0.0017	0.3197±0.0183	0.3556±0.0379
Ave. Rank	**1.0769**	4.0000	3.8462	4.0000	4.4615	3.6154

cluster, thus intuitively demonstrating the autonomous k selection ability of FedCCL.

Convergence Evaluation: To evaluate the efficiency in convergence of FedCCL, we plot the values of the objective function of FedCCL on two datasets in Fig. 5. It can be observed that FedCCL converges quickly within 50 iterations in most cases. Moreover, the objective function always experiences a steep decline after the server updates, confirming that the designed server seeds interaction mechanism is highly effective. It is also noteworthy that, since only limited statistics are permitted to be communicated between clients and server, and the data distribution varies on the clients and server, the convergence curve in Fig. 5 is not monotonically decreasing. Such an effect is reasonable because the learning objective can be viewed as heterogeneous at different clients and the server.

Clustering Performance Evaluation: To further validate the effectiveness of FedCCL, we have also compared it with the existing approaches in the challenging non-IID scenario. Specifically, we generate 20 different sets of data distributions for clients using k-means for each dataset. The number of clients is uniformly set to 5 in this comparison and also in the following ablation study experiments. As FedCCL does not require the 'true' cluster number k^*, we randomly select the k from the range $[k^*, 2k^*]$ as the initial cluster number. The performance in terms of SC index is shown in Table III. Due to space limitation, results w.r.t. CH index is omitted here. The best and the second-best results are highlighted using boldface and underline, respectively. The 'Ave. Rank' row reports the average ranks of different counterparts across all datasets.

It can be observed from Table III that FedCCL outperforms the other counterparts in general, indicating its effectiveness. Specifically, FedCCL surpasses the counterparts on almost all the datasets, except for the SP dataset where it still achieves the second-best result. The above observations illustrate that the proposed method can effectively find the global optimal position of centroids by maximizing both the intra-cluster density and the inter-cluster dispersion.

Ablation Study: We conduct ablation study by using the discriminative CH index to validate the effectiveness of the core server competition process. Due to space limitation, results w.r.t. SC index is omitted here. The version of FedCCL without ServerSI (w/o ServerSI) is formed by replacing the ServerSI with the simple cluster centroid aggregation of FFCM. That is, the global centroid is computed as a weighted average of the centroids from clients, where the weight assigned to each client's centroid is proportional to the number of objects it represents. It can be observed from Table II that FedCCL outperforms w/o ServerSI in most cases. This intuitively illustrates the effectiveness of ServerSI, which can let the seed points sufficiently interact with the neighboring ones. As a result, the detailed local distribution information can be iteratively propagated across different seed points to collaboratively eliminate redundant seeds and sketch the global cluster distributions.

V. CONCLUSION

This paper has proposed a new federated clustering approach called FedCCL that can well mine global cluster distributions upon heterogeneous data distributions of clients. It advances federated clustering to a more challenging but realistic scenario, i.e., all the clients can be extremely non-IID and the 'true' number of clusters of the clients and the server are all unknown. More specifically, FedCCL assigns excessive seed points to the clients to outline their local distribution using aggregated update intensity of seed points received locally. To address the potential contradiction among seeds caused by the clients' heterogeneity, interactions across both clients and seeds have been facilitated to more comprehensively explore clusters. FedCCL is easy to use as it is robust to an easy-to-set learning rate. Comprehensive experiments have been conducted to illustrate its efficacy.

Despite the effectiveness of FedCCL, there are still some noteworthy limitations. That is, we assume static federated clustering on pure numerical data, and thus FedCCL is incompetent to asynchronous updates of clients. The next promising avenue would involve dynamic federated clustering of datasets described by a mixture of numerical and categorical attributes.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants: 62102097,

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.

62376233, and 62306181, NSFC/Research Grants Council (RGC) Joint Research Scheme under grant N_HKBU214/21, General Research Fund of RGC under grants: 12201321, 12202622, and 12201323, Natural Science Foundation of Guangdong Province under grants: 2023A1515012855 and 2024A1515010163, RGC Senior Research Fellow Scheme under grant SRFS2324-2S02, Shenzhen Science and Technology Program under grant RCBS20231211090659101, and the Xiaomi Young Talents program.

References

[1] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, "Federated learning review: Fundamentals, enabling technologies, and future applications," *Information Processing & Management*, vol. 59, no. 6, p. 103061, 2022.

[2] X. Yin, Y. Zhu, and J. Hu, "A Comprehensive Survey of Privacy-preserving Federated Learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–36, 2022.

[3] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, p. 106775, 2021.

[4] E. Lubana, C. I. Tang, F. Kawsar, R. Dick, and A. Mathur, "Orchestra: Unsupervised Federated Learning via Globally Consistent Clustering," in *ICML*, vol. 162, 2022, pp. 14 461–14 484.

[5] A. Nelus, R. Glitza, and R. Martin, "Unsupervised Clustered Federated Learning in Complex Multi-source Acoustic Environments," in *EUSIPCO*, 2021, pp. 1115–1119.

[6] S. Xie, Y. Wu, K. Liao, L. Chen, C. Liu, H. Shen, M. Tang, and L. Sun, "Fed-SC: One-Shot Federated Subspace Clustering over High-Dimensional Data," in *ICDE*, 2023, pp. 2905–2918.

[7] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An Efficient Framework for Clustered Federated Learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 19 586–19 597.

[8] D. K. Dennis, T. Li, and V. Smith, "Heterogeneity for the Win: One-Shot Federated Clustering," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 2611–2620.

[9] A. Kumar and R. Kannan, "Clustering with Spectral Norm and the k-Means Algorithm," in *FOCS*, 2010, pp. 299–308.

[10] P. Awasthi and O. Sheffet, "Improved Spectral-Norm Bounds for Clustering," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2012, pp. 37–49.

[11] W. Pedrycz, "Federated FCM: Clustering Under Privacy Requirements," *IEEE Transactions on Fuzzy Systems*, pp. 3384–3388, 2022.

[12] S. Morris and W. Anna, "Towards Federated Clustering: A Federated Fuzzy c-Means Algorithm (FFCM)," 2022.

[13] Y.-M. Cheung, "A Competitive and Cooperative Learning Approach to Robust Data Clustering," in *Neural Networks and Computational Intelligence - 2004*, 2004, pp. 131–136.

[14] L.-T. Law and Y.-M. Cheung, "Color image segmentation using rival penalized controlled competitive learning," in *IJCNN*, 2003, pp. 108–112.

[15] S. Ding, C. Li, X. Xu, L. Guo, L. Ding, and X. Wu, "Horizontal Federated Density Peaks Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2023.

[16] Y.-M. Cheung, "Rival penalization controlled competitive learning for data clustering with unknown cluster number," in *ICONIP*, vol. 1, 2002, pp. 467–471.

[17] Y.-M. Cheung and J. Hong, "Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number," *Pattern Recognition*, vol. 46, no. 8, pp. 2228–2238, 2013.

[18] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," *SODA*, pp. 1027–1035, 2007.

[19] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984.

[20] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[21] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable K-Means+," *VLDB*, vol. 5, no. 7, 2012.

[22] A. Asuncion and D. Newman, "UCI machine learning repository," 2007.

[23] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[24] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[25] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

Authorized licensed use limited to: Guangdong Univ of Tech. Downloaded on December 12,2024 at 02:23:26 UTC from IEEE Xplore. Restrictions apply.