A Convergent Federated Clustering Algorithm without Initial Condition

Harsh Vardhan [1] **Avishek Ghosh** [2] **Arya Mazumdar** [1]

Abstract

In this paper, we define a new clustering framework for FL based on the (optimal) local models of the users: two users belong to the same cluster if their local models are close. We propose an algorithm, *Successive Refine Federated Clustering Algorithm* (SR-FCA), that treats each user as a singleton cluster as an initialization, and then successively refine the cluster estimation via exploiting similarity with other users. In any intermediate step, SR-FCA uses an *error-tolerant* federated learning algorithm within each cluster to exploit simultaneous training and to correct clustering errors. Unlike some prominent prior works, such as (Ghosh et al., 2022), SR-FCA does not require any *good* initialization (or warm start), both in theory and practice. We show that with proper choice of learning rate, SR-FCA incurs arbitrarily small clustering error. Additionally, SR-FCA does not require the knowledge of the number of clusters apriori like some prior works. We also validate the performance of our algorithm on real-world FL datasets including FEMNIST and Shakespeare in non-convex problems and show the benefits of SR-FCA over several baselines.

1. Introduction

The issue of heterogeneity is crucial for FL, since the data resides in users' own devices, and naturally no two devices have identical data distribution. There has been a rich body of literature in FL to address this problem of non-iid data. We direct the readers to two survey papers (and the references therein), (Li et al., 2020; Kairouz et al., 2019) for a comprehensive list of papers on heterogeneity in FL. A line of research assumes the *degree of dissimilarity* across users is small, and hence focuses on learning a single global model (Zhao et al., 2018; Sahu et al., 2018a; Li et al., 2018; Sattler et al., 2019b; Mohri et al., 2019; Karimireddy et al., 2020). Another line of research in FL focuses on obtaining models personalized to individual users. For example (Sahu et al., 2018b; Li et al., 2021) uses a regularization to obtaining individual models for users and the regularization ensures that the local models stay close to the global model. Another set of work poses the heterogeneous FL as a meta learning problem (Chen et al., 2018; Jiang et al., 2019; Fallah et al., 2020b;a). Here, the objective is to first obtain a single global model, and then each device run some local iterations (fine tune) the global model to obtain their local models. Furthermore (Collins et al., 2021) exploits shared representation across users by running an alternating minimization algorithm and personalization. Note that all these personalization algorithms, including meta learning, work only when the local models of the users' are close to one another (see bounded heterogeneity terms γ_H and γ_G terms in Assumption 5 of (Fallah et al., 2020b)).

On the other spectrum, when the local models of the users may not be close to one another, (Sattler et al., 2019a; Mansour et al., 2020; Ghosh et al., 2022) propose a framework of *Clustered Federated Learning*. Here users with dissimilar data are put into different clusters, and the objective is to obtain individual models for each cluster; i.e., a joint training is performed within each cluster. Among these, (Sattler et al., 2019a) uses a top-down approach using cosine similarity metric between gradient norm as optimization objective. However, it uses a centralized clustering scheme, where the center has a significant amount of compute load, which is not desirable for FL. Also, the theoretical guarantees of (Sattler et al., 2019a) are limited. Further, in (Duan et al., 2020), a data-driven similarity metric is used extending the cosine similarity and the framework of (Sattler et al., 2019a). Moreover, in (Mansour et al., 2020), the authors propose algorithms for both clustering and personalization. However, they provide guarantees only on generalization, not iterate convergence. In (Smith et al., 2017) the job of multi-task learning is framed as clustering where a regularizer in the optimization problem defines clustering objective.

Very recently, in (Ghosh et al., 2022), an iterative method in the clustered federated learning framework called Iterative Federated Clustering Algorithm, or IFCA, was proposed and a *local convergence* guarantee was obtained. The problem setup for IFCA is somewhat restrictive—it requires the

[1] Halicioglu Data Science Institute University of California, San Diego, USA. [2] Centre for Machine Intelligence and Data Science, Indian Institute of Technology, Bombay, India.. Correspondence to: Harsh Vardhan <hharshvardhan@ucsd.edu>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. Copyright 2023 by the author(s).

model (or data distribution) of all the users in the same cluster to be (exactly) identical. IFCA alternately estimates the cluster identities of the users and optimizes model parameters for the user clusters via gradient descent. In order to converge, IFCA necessarily requires *suitable* initialization in clustering, which can be impractical. Furthermore, in (Ghosh et al., 2022), all the users are partitioned into a fixed and known number of clusters, and it is discussed in the same paper that the knowledge about the number of clusters is quite non-trivial to obtain (see Section 6.3 in (Ghosh et al., 2022)).

Following IFCA, a number of papers attempt to extend the federated clustering framework (Ruan & Joe-Wong, 2021; Xie et al., 2020), however, the crucial shortcomings of IFCA, namely the requirements on *good* initialization and *identical* local models still remain unaddressed to the best of our knowledge.

In this paper, we address the above-mentioned shortcomings. We introduce a new clustering algorithm, Successive Refinement Federated Clustering Algorithm or SR-FCA, which leverages pairwise distance based clustering and refines the estimates over multiple rounds. We show that SR-FCA does not require any specific initialization. Moreover, we can allow the same users in a cluster to have non-identical models (or data distributions); only certain degree of similarity is sufficient. In Section 2 we define a novel clustering structure (see Definition 2.1), which allows the the local models of the users to be different (we denote this discrepancy by parameter $\epsilon_1(\geq 0)$, and for IFCA, $\epsilon_1 = 0$). Furthermore, SR-FCA works with a different set of hyper-parameters which does not include the number of clusters and SR-FCA iteratively estimates this hyper-parameter.

Clustering Framework and Distance Metric: Classically, clustering is defined in terms of distribution from which the users sample data. However, in a federated framework, it is common to define a heterogeneous framework such as clustering in terms of other discrepancy metric; for example in (Mansour et al., 2020), a metric that depends on the local loss is used. In this paper, we use a distance metric across users' local model as a discrepancy measure and define a clustering setup based on this. Our distance metric may in general include non-trivial metric like Wasserstein distance, ℓ_q norm (with $q \geq 1$) that captures desired practical properties like permutation invariance and sparsity for (deep) neural-net training. For our theoretical results, we focus on strongly convex and smooth loss for which ℓ_2 norm of iterates turns out to be the natural choice. However, for non-convex neural networks on which we run most of our experiments, we use a *cross-cluster loss* metric. For two clients i,j, we define their cross-cluster loss metric as the average of the loss of one client on the other's model, i.e., client i's loss on the model of j and the other way round. If this metric is low, we can use the model of client i for client j and vice-versa, implying that the clients are similar. We explain this in detail in Appendix C. With the above discrepancy metric, we put the users in same cluster if their local models are close – otherwise they are in different clusters. Under suitable assumptions, we provide theoretical guarantees on SR-FCA in Section 4. Further, using the cross-cluster loss metric in experiments, we show that SR-FCA outperforms all baselines (including IFCA) for real datasets.

2. Federated Clustering and Our Setup

In this section, we formally define the clustering problem. Let, $[n] \equiv \{1,2,...,n\}$. We have m users (or machines) that are partitioned into disjoint clusters, denoted by the clustering map $\mathcal{C}^\star : [m] \to [C]$, where C is the (unknown) number of clusters. Each user $i \in [m]$ contains $n_i \geq n$ data points $\{z_{i,j}\}_{j=1}^{n_i}$ sampled from a distribution $\mathcal{D}_i$. We define $f(\cdot;z) : \mathcal{W} \to \mathbb{R}$ as the loss function for the sample z, where $\mathcal{W} \subseteq \mathbb{R}^d$. Here, $\mathcal{W}$ is a closed and convex set with diameter D. We now define the population loss, $F_i : \mathcal{W} \to \mathbb{R}^d$, and its minimizer, $w_i^\star$, for each user $i \in [m]$: $F_i(w) = \mathbb{E}_{z\sim\mathcal{D}_i}[f(w,z)]$, $\quad w_i^\star = \min_{w\in\mathcal{W}} F_i(w)$. The original clustering $\mathcal{C}^\star$ is based on the population minimizers of users, $w_i^\star$. This is defined as:

Definition 2.1 (Clustering Structure)**.** For a distance metric $\mathsf{dist}(.,.)$ with non-negative constants ϵ_1,ϵ_2 with $\epsilon_2 > \epsilon_1$, the local models satisfy

$$\max_{\mathcal{C}^\star(i)=\mathcal{C}^\star(j)} \mathsf{dist}(w_i^\star, w_j^\star) \leq \epsilon_1,$$
$$\min_{\mathcal{C}^\star(i)\neq\mathcal{C}^\star(j)} \mathsf{dist}(w_i^\star, w_j^\star) \geq \epsilon_2$$

The above structure is illustrated in Figure 1. This allows the population minimizers inside clusters to be close, but not necessarily equal, as opposed to (Ghosh et al., 2022)(i.e., IFCA assumes $\epsilon_1 = 0$). We emphasize that $\epsilon_1 \neq 0$ is more practical, as similar users may have close but never identical local models.

In practice, we have access to neither F_i nor $w_i^\star$, but only the sample mean variant of the loss, the empirical risk, $f_i(w) = \frac{1}{n_i}\sum_{j=1}^{n_i} f(w, z_{i,j})$ for each user $i \in [m]$. Let $G_c \equiv \{i : i \in [m], \mathcal{C}^\star(i) = c\}$ denote the set of users in cluster c according to the original clustering $\mathcal{C}^\star$. We can then define the population loss and its minimizer, per cluster $c \in [C]$ as $\mathcal{F}_c$

$$\mathcal{F}_c(w) = \frac{1}{|G_c|}\sum_{i\in G_c} F_i(w), \quad \omega_c^* = \operatorname*{argmin}_{w\in\mathcal{W}} \mathcal{F}_c(w) \qquad (1)$$

Our final goal is to find a population loss minimizer for each cluster $c \in [C]$, i.e., ω_c^*. To obtain this, we need to find the correct clustering $\mathcal{C}^\star$ and recover the minimizer of each cluster's population loss. There are two major difficulties in this setting: (a) the number of clusters is not known beforehand. This prevents us from using most clustering algorithms like k-means, and (b) The clustering depends on $w_i^\star$ which we do not have access to. We can estimate $w_i^\star$ by minimizing f_i, however when n, the minimum number of data points per user, is small, this estimate may be very far from $w_i^\star$. These difficulties can be overcome by utilizing clustered federated learning, which we describe in the next section.

3. Algorithm : SR-FCA

In this section, we formally present our clustering algorithm, SR-FCA. We first run the subroutine ONE_SHOT to obtain an appropriate initial clustering, which can be further improved. SR-FCA then successively calls the REFINE() subroutine to improve this clustering. In each step of REFINE(), we first estimate the cluster models for each cluster. Then, based on these models we regroup all the users using RECLUSTER() and, if required, we merge the resulting clusters, using MERGE().

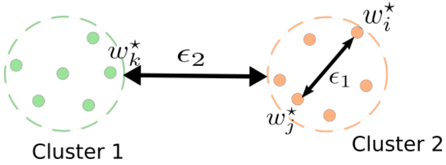

Figure 1: The dots represent the population risk minimizers for two clusters in dist(.,.) space according to $\mathcal{C}^\star$.

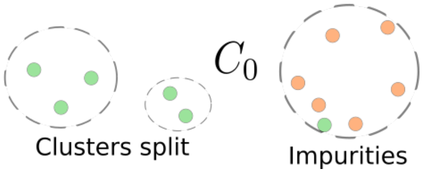

Figure 2: The dots represent the ERM in dist(.,.) space and the corresponding clustering $\mathcal{C}_0$ obtained after ONE_SHOT

ONE_SHOT() For our initial clustering, we create edges between nodes based on the distance between their locally trained models if $\mathsf{dist}(w_i, w_j) \leq \lambda$, for a threshold λ and then obtain clusters from this graph by correlation clustering (Bansal et al., 2002), which is a graph clustering that minimizes total number of edges across the clusters, and non-edges within the clusters (Bansal et al., 2002). We only keep the clusters which have at least t nodes.

If our locally trained models, $w_{i,T}$, were close to their population minimizers, $w_i^\star$, for all nodes $i \in [m]$, then choosing a threshold $\lambda \in (\epsilon_1, \epsilon_2)$, we obtain edges between only clients which were in the same cluster in $\mathcal{C}^\star$. However, if n, the number of local datapoints is small, then our estimates of local models $w_{i,T}$ might be far from their corresponding $w_i^\star$ and we will not be able to recover $\mathcal{C}^\star$.

However, $\mathcal{C}_0$ is still a good clustering if it satisfies these requirements: (a) if every cluster in the range of the clustering map $\mathrm{rg}(\mathcal{C}^\star) = [C]$ has a good proxy (in the sense of Definition 3.1) in $\mathrm{rg}(\mathcal{C}_0)$, and (b) each cluster in $\mathrm{rg}(\mathcal{C}_0)$ has at most a small fraction ($< \frac{1}{2}$) of mis-clustered users in it. E.g., Figure 2 provides an example of one such good clustering when $\mathcal{C}^\star$ is defined according to Figure 1. We can see that even though $\mathcal{C}_0 \neq \mathcal{C}^\star$, the two green clusters and the single orange cluster in $\mathcal{C}_0$ are mostly "pure" and are proxies of Cluster 1 and Cluster 2 in Figure 1.

Algorithm 1 SR-FCA

```
Input: Threshold $\lambda$, Size parameter $t$
Output: Clustering $C_R$
$C_0 \leftarrow$ ONE_SHOT($\lambda, t$)
for $r = 1$ to $R$ do
    $C_r \leftarrow$ REFINE($C_{r-1}, \lambda$)
end for
ONE_SHOT($\lambda, t$)
for all $i$ clients in parallel do
    $w_{i,T} \leftarrow$ Train local model for client $i$ for $T$ steps
end for
$G \leftarrow$ Graph with $m$ vertices and no edges
for all pairs of clients $i, j \in [m], i \neq j$ do
    Add edge $(i,j)$ to the graph $G$ if $\mathsf{dist}(w_{i,T}, w_{j,T}) \leq \lambda$
end for
$C_0 \leftarrow$ Clusters from graph $G$ with size $\geq t$
by correlation clustering of (Bansal et al., 2002).
REFINE($C_{r-1}, \lambda$)
for all clusters $c \in C_{r-1}$ do
    $\omega_{c,T} \leftarrow$ TrimmedMeanGD()
end for
$C_r' \leftarrow$ RECLUSTER($C_{r-1}$)
$C_r \leftarrow$ MERGE($C_r', \lambda, t$)
```

To formally define the notion of "purity" and "proxy", we introduce the notion of cluster label for any arbitrary clustering $\mathcal{C}'$, which relates it to the original clustering $\mathcal{C}^\star$.

Definition 3.1 (Cluster label). We define $c \in [C]$, as the cluster label of cluster $c' \in \mathrm{rg}(C')$ if the majority ($> 1/2$ fraction) of nodes in c' are originally from c.

This definition allows us to map each cluster $c' \in \mathrm{rg}(C')$ to a cluster c in $\mathcal{C}^\star$ and thus define the notion of "proxy". In Figure 2, the cluster label of green clusters is Cluster 1 and that of orange cluster is Cluster 2. Further, using the cluster label c, we can define the impurities in cluster c' as the nodes which did not come from c'. In Figure 2, the green node in orange cluster is an impurity. Based on these definitions, we can see that if clusters in $\mathcal{C}_0$ are mostly pure and can represent all clusters in $\mathcal{C}^\star$, then $\mathcal{C}_0$ is a good clustering.

Subroutine TrimmedMeanGD(): The main issue with ONE_SHOT(), namely, small n, can be mitigated if we use federation. Since, $\mathcal{C}_0$ has atleast t nodes per cluster, training a single model for each cluster will utilize $\geq tn$ datapoints, making the estimation more accurate.

However, from Figure 2, we can see that the clusters contain impurities, i.e., users from a different cluster. To handle them, we use a robust training algorithm, TrimmedMean(Yin et al., 2018). This subroutine is similar to FedAvg (McMahan et al., 2016), but instead of taking the average of local models, we take the coordinate-wise trimmed mean, where $\beta \in (0,1/2)$ defines the trimming level.

The full algorithm for `TrimmedMeanGD` and the definition of trimmed mean is provided in Appendix A. Note that coordinate-wise trimmed mean has been used to handle Byzantine users, achieving optimal statistical rates (Yin et al., 2018), when $< \beta$ fraction of the users are Byzantine. For our problem setting, users from different clusters are treated as impurities.

Note the two requirements for good clustering $\mathcal{C}_0$ from `ONE_SHOT`: (a) if every cluster in $\mathcal{C}^\star$ has a proxy in $\mathcal{C}_0$, then the `TrimmedMeanGD` obtains at least one cluster model for every cluster in $\mathcal{C}^\star$, (b) if every cluster in $\mathcal{C}_0$ has a small fraction ($\beta < \frac{1}{2}$) of impurities, then trimmed mean can recover the correct cluster model for every cluster.

We end up with a trained model for each cluster as an output of this subroutine. Since these models are better estimates of their population risk minimizers than before, we can use them to improve $\mathcal{C}_0$.

Subroutine `RECLUSTER()`: The full algorithm for this subroutine is provided in Algorithm 3. This subroutine reduces the impurity level of each cluster in $\mathcal{C}_0$ by assigning each client i to its nearest cluster c in terms of $\mathsf{dist}(\omega_{c,T}, w_{i,T})$. Since $\omega_{c,T}$ are better estimates, we hope that the each impure user will go to a cluster with its actual cluster label. For instance, in Figure 2, the impure green node should go to one of the green clusters. If some clusters in $\mathrm{rg}(\mathcal{C}^\star)$ does not have a good proxy in $\mathrm{rg}(\mathcal{C}_0)$, then the nodes of this cluster will remain as impurities.

Subroutine `MERGE()`: We provide the full algorithm for this subroutine in Algorithm 4. Even after removing all impurities from each cluster, we can still end up with clusters in $\mathcal{C}^\star$ being split, for instance the green clusters in Figure 2. In $\mathcal{C}^\star$, these form the same cluster, thus they should be merged. As these were originally from the same cluster in $\mathcal{C}^\star$, their learned models should be very close should thus be merged. Similar to `ONE_SHOT`, we create a graph G but instead with vertex set being the clusters in $\mathcal{C}'_r$. Then, we add edges between clusters based on a threshold λ and find all the clusters in the resultant graph G by correlation clustering. Then, each of these clusters in G correspond to a set of clusters in $\mathcal{C}'_r$, so we merge them into a single cluster to obtain the final clustering $\mathcal{C}_{r+1}$.

4. Theoretical Guarantees

In this section, we present a brief overview of our theoretical results. For theoretical tractability, we impose additional conditions on `SR-FCA`. First, the $\mathsf{dist}(.,.)$ is the euclidean (ℓ_2) norm. Note that for realizable linear regression, ℓ_2 norm is the appropriate choice of distance metric (see Proposition B.1). Further, the hyperparameters have the following requirements : $\lambda \in (\epsilon_1, \epsilon_2)$ and $t \leq c_{\min}$, where $c_{\min}$ is the minimum size of the cluster. Although in Algorithm 1, we use correlation clustering for finding clusters from a graph, in theory, we restrict ourselves to finding cliques only. We specify the exact class of loss functions for which our analysis holds defined by the following assumptions, with definitions provided in Appendix I. We specify the exact class of loss functions for which our analysis holds by the following assumptions, with definitions provided in Appendix I.

Assumption 4.1 (Strong convexity)**.** The loss per sample $f(w,.)$ is μ-strongly convex with respect to w.

Assumption 4.2 (Smoothness)**.** The loss per sample $f(w,.)$ is also L-smooth with respect to w.

Assumption 4.3 (Lipschitz)**.** The loss per sample $f(w,.)$ is L_k-Lipschitz for every coordinate $k \in [d]$. Define $\hat{L} = \sqrt{\sum_{k=1}^{d} L_k^2}$.

We want to emphasize that the above assumptions are standard and have appeared in the previous literature. For example, the strong convexity and smoothness conditions are often required to obtain theoretical guarantees for clustering models (see (Ghosh et al., 2022; Lu & Zhou, 2016), which includes `IFCA` and the classical k-means which assume a quadratic objective. The coordinate-wise Lipschitz assumption is also not new and (equivalent assumptions) featured in previous works (see (Yin et al., 2018; 2019), with it being necessary to establish convergence of the trimmed mean procedure.

Throughout this section, we assume the above assumptions hold. We provide guarantees on the misclustering error and the convergence of iterates of `SR-FCA`.

To measure misclustering error of `SR-FCA`, we quantify the probability of not recovering the original clustering, i.e., $\mathcal{C}_r \neq \mathcal{C}^\star$. We provide the guarantees for `ONE_SHOT` and a single-step of `REFINE`.

Theorem 4.4 (Error after `ONE_SHOT`)**.** *After running `ONE_SHOT` with $\eta \leq \frac{1}{L}$ for T iterations, for the threshold $\lambda \in (\epsilon_1, \epsilon_2)$ and some constant $b_2 > 0$, the probability of error is $\Pr[C_0 \neq C^\star] \leq p \equiv md \ \exp(-n\frac{b_2 \Delta}{\hat{L}\sqrt{d}})$, provided $\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}} \lesssim d$, where $\Delta = \frac{\mu}{2}(\frac{\min\{\epsilon_2 - \lambda, \lambda - \epsilon_1\}}{2} - (1 - \frac{\mu}{L})^{T/2} D)$ and $n = \min_{i \in [m]} n_i$.*

Theorem 4.5 (One step `REFINE()`)**.** *Let $\beta t = \Theta(c_{\min})$ where `REFINE()` is run with `TrimmedMeanGD`(β) with $\eta \leq \frac{1}{L}$. Provided $\min\{\frac{n^{2/3}\Delta'^{4/3}}{D^{2/3}}, \frac{n^2 \Delta'^2}{\hat{L}^2 \log(c_{\min})}\} \gtrsim d$, with $0 < \beta < \frac{1}{2}$, where $\Delta' = \Delta - \frac{\mu B}{2} > 0$ and $B = \sqrt{2\hat{L}\epsilon_1/\mu} << \frac{2\Delta'}{\mu}$, for large m and n, after running 1 step of `REFINE`, we have, $\Pr[C_1 \neq C^\star] \leq \frac{\rho_1}{m^{1-\rho_2}} p$ for some small constants $\rho_1 > 0$ and $\rho_2 \in (0,1)$.*

We would like to emphasize that the probability of error is exponential in both n and the separation Δ, yielding a *reasonable* good clustering after the ONE_SHOT step. Note that the best probability of error is obtained when $\lambda = \frac{\epsilon_1 + \epsilon_2}{2}$. Further, we require separation $\Delta = \Omega(\frac{\log m}{n})$ for $p < 1$. Additionally, a single step of REFINE() brings down the misclustering error by a factor of $\frac{1}{m}$. Note that for IFCA, $\epsilon_1 = 0$, so the condition $B << \frac{\Delta'}{2\mu}$ is automatically satisfied. Assuming that the datapoints on each machine are resampled at every REFINE step, the improvement in a single REFINE step can be extrapolated to R REFINE steps. This result is deferred to Appendix B.

We also obtain an appropriate loss minimizer for each cluster.

Theorem 4.6 (Cluster iterates)**.** *Under the conditions described in Theorem 4.5, after running* SR-FCA *for* $(R+1)$ *steps of* REFINE()*, where on each machine* $i \in [m]$, n_i *datapoints are resampled and* w_i *is recomputed as in* ONE_SHOT *at every* REFINE *step, we have* $C^{R+1} = C^\star$ *and*

$$\|\omega_{c,T} - \omega_c^\star\| \leq (1-\kappa^{-1})^{T/2} D + \Lambda + 2B,$$

$$\textit{where, } \Lambda = \mathcal{O}\Big(\frac{\hat{L}d}{1-2\beta}\Big(\frac{\beta}{\sqrt{n}} + \frac{1}{\sqrt{nc_{\min}}}\Big)\sqrt{\log(nm\hat{L}D)}\Big)$$

$\forall c \in \mathrm{rg}(C^\star)$, *with probability* $1 - \big(\frac{\rho_2}{m^{(1-\rho_1)}} p\big)^R - \frac{m}{c_{\min}} \frac{4du''}{(1+nc_{\min}\hat{L}D)^d}$*, for some constant* $u'' > 0$*.*

The iterates converge exponentially fast to the true cluster minima $\omega_c^\star$, which matches that of IFCA. Additional theoretical details are deferred to Appendix B.

5. Experiments

Setup. We compare the empirical performance of SR-FCA against several baselines on real and simulated datasets. The results for simulated datasets are deferred to Appendix C. The real datasets are FEMINST and Shakespeare from leaf database (Caldas et al., 2018). We compare with standard FL baselines – Local (every client trains its own local model) and Global (a single model trained via FedAvg on all clients). The main baseline we compare to is IFCA. Among clustered FL baselines, we consider CFL (Sattler et al., 2019a), Local-KMeans (Ghosh et al., 2019) (KMeans on the model weights of each client's local model), FedSoft (Ruan & Joe-Wong, 2021) (IFCA with soft clustering) and ONE_SHOT-IFCA (initial clustering of IFCA obtained by ONE_SHOT), to assess if these variants can fix the issues of initialization in IFCA. For SR-FCA, we tune the parameters λ and β for trimmed mean and set $t = 2$ and require at most 2 REFINE steps. Further, for clustered FL baselines (IFCA, we tune the number of clusters.

Distance Metric We utilize a novel distance metric based on cross-cluster loss which is better suited to measure distances between clients' models as these are neural networks.

Definition 5.1 (Cross-Cluster distance)**.** For any two clients $i,j \in [m]$, with corresponding local models w_i and w_j and local empirical losses f_i and f_j, we define the cross cluster loss metric as

$$\mathsf{dist}_{\text{cross-cluster}}(w_i, w_j) = \frac{1}{2}(f_i(w_j) + f_j(w_i))$$

Table 1: Test Accuracy and standard deviations across 5 seeds on Real datasets. The highest accuracy is **bold**. SR-FCA consistently outperforms all baselines.

BASELINE	FEMNIST	SHAKESPEARE
SR-FCA	**83.83**± 1.49	**48.54 ± 0.69**
LOCAL	66.18 ± 2.14	33.86 ±1.22
GLOBAL	80.00± 3.02	45.28 ± 0.78
CFL	79.48 ± 3.48	44.14 ± 1.03
IFCA	81.93± 1.56	46.12 ± 1.22
FEDSOFT	78.74 ± 2.61	46.98 ± 1.25
ONE_SHOT-IFCA	81.62 ± 2.29	45.56 ± 1.15

To extend this definition to distances between cluster c and client j, such as those required by REFINE, we replace the client model w_i and client loss f_i by the cluster model w_c and empirical cluster loss f_c respectively. Similarly, to obtain distances between clusters c and c', which are required by MERGE, we replace the client models and losses by cluster models and losses respectively.

As the true clustering is not known for real datasets, we report only the final test accuracy in Table 1.

5.1. Results

Across all datasets, we find that SR-FCA outperforms all the baselines. The Local algorithm has access to little data, while the Global model cannot handle the heterogeneity. Hence, most clustered FL baselines outperform them . CFL and Local-KMeans use the cosine distance between gradients and l_2 distance between model weights which are not suitable for NN models. Note that we do not report the test accuracy for Local-KMeans as it is $\leq 5\%$. Therefore, both IFCA and its variants and SR-FCA outperform them.

Comparison with IFCA: On real datasets, $\epsilon_1 \neq 0$ as the clients inside a cluster may be close but not identical. In addition to this, for IFCA, we need to find the correct number of clusters by tuning. For a random sample of clients, the true number of clusters might not be the same. SR-FCA can compute both the correct clustering and cluster iterates for every random sample, allowing it to beat IFCA, which fits the same number of clusters to every random sample. The difference is more pronounced for the more difficult Shakespeare dataset than the easier FEMNIST dataset. Further, the variants of IFCA – FedSoft and ONE_SHOT-IFCA, have similar test performance to IFCA. For FedSoft, which is a soft-clustering version of IFCA, the issue of initialization remains unresolved. Running IFCA after ONE_SHOT can only re-cluster the clients thus results in a similar performance. In short, SR-FCA outperforms IFCA as well as its variants.

References

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. In *Machine Learning*, pp. 238–247, 2002.

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018. URL http://arxiv.org/abs/1812.01097.

Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Duan, M., Liu, D., Ji, X., Liu, R., Liang, L., Chen, X., and Tan, Y. Fedgroup: Efficient clustered federated learning via decomposed data-driven measure. *arXiv preprint arXiv:2010.06870*, 2020.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

Fallah, A., Mokhtari, A., and Ozdaglar, A. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092. PMLR, 2020a.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized Federated Learning: A Meta-Learning Approach. *arXiv:2002.07948 [cs, math, stat]*, October 2020b. URL http://arxiv.org/abs/2002.07948. arXiv: 2002.07948.

Ghosh, A., Hong, J., Yin, D., and Ramchandran, K. Robust federated learning in a heterogeneous environment. *CoRR*, abs/1906.06629, 2019. URL http://arxiv.org/abs/1906.06629.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12):8076–8091, 2022. shorter version in NeurIPS 2021.

Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132–5143. PMLR, November 2020. URL https://proceedings.mlr.press/v119/karimireddy20a.html. ISSN: 2640-3498.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Li, L., Xu, W., Chen, T., Giannakis, G. B., and Ling, Q. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. *arXiv preprint arXiv:1811.03761*, 2018.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021.

Lloyd, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

Lu, Y. and Zhou, H. H. Statistical and computational guarantees of lloyd's algorithm and its variants. *arXiv preprint arXiv:1612.02099*, 2016.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Aguera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Page, D. How to Train your ResNet 4 : Architecture. https://myrtle.ai/learn/how-to-train-your-resnet-4-architecture/, 2019.

Ruan, Y. and Joe-Wong, C. Fedsoft: Soft clustered federated learning with proximal local updating. *CoRR*, abs/2112.06053, 2021. URL https://arxiv.org/abs/2112.06053.

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3, 2018a.

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A. S., and Smith, V. On the convergence of federated optimization in heterogeneous networks. *ArXiv*, abs/1812.06127, 2018b.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. *arXiv preprint arXiv:1910.01991*, 2019a.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2019b.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/6211080fa89981f66b1a0c9d55c61d0f-Paper.pdf.

Xie, M., Long, G., Shen, T., Zhou, T., Wang, X., and Jiang, J. Multi-center federated learning. *CoRR*, abs/2005.01026, 2020. URL https://arxiv.org/abs/2005.01026.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Defending against saddle point attack in Byzantine-robust distributed learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7074–7084. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yin19a.html.

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

Appendix for "A Convergent Federated Clustering Algorithm without Initial Condition"

A. Additional Description of SR-FCA

SR-FCA uses a bottom-up approach to construct and refine clusters. The initialization in ONE_SHOT is obtained by distance-based thresholding on local models. These local models are improper estimates of their population minimizers due to small n, causing $\mathcal{C}_0 \neq \mathcal{C}^\star$. However, if $\mathcal{C}_0$ is not very bad, i.e., each cluster has $< \frac{1}{2}$ impurity fraction and all clusters in $\mathcal{C}^\star$ are represented, we can refine it. REFINE() is an alternating procedure, where we first estimate cluster centers from impure clusters. Then, we RECLUSTER() to remove the impurities in each cluster and then MERGE() the clusters which should be merged according to $\mathcal{C}^\star$. Note that as these steps use cluster estimates which are more accurate, they should have smaller error. This iterative procedure should recover one cluster for each cluster in $\mathcal{C}^\star$, thus obtaining the number of clusters and every cluster should be pure, so that $\mathcal{C}^\star$ is exactly recovered. Note that the TrimmedMeanGD procedure also returns trained models, however, these may not have the best performance. Once we have recovered $\mathcal{C}^\star$, we can run a FL algorithm inside each cluster if we need better cluster models. The analysis of computation and communication complexity of SR-FCA is deferred to Appendix D.

We provide the full algorithms for TrimmedMeanGD, RECLUSTER and MERGE in Algorithm 2,Algorithm 3 and Algorithm 4 respectively. Further, we provide the definition of the coordinate-wise trimmed mean operation in Definition A.1.

Algorithm 2 TrimmedMeanGD()

```
Input: $0 \leq \beta < \frac{1}{2}$, Clustering $\mathcal{C}_r$
Output: Cluster models $\{\omega_{c,T}\}_{c \in \mathrm{rg}(\mathcal{C}_r)}$
for all clusters $c \in \mathrm{rg}(\mathcal{C}_r)$ in parallel do
    $w_{c,0} \leftarrow w_0$
    for $t = 0$ to $T-1$ do
        $g(w_{c,t}) \leftarrow \mathrm{TrMean}_\beta(\{\nabla f_i(w_{c,t}), \mathcal{C}_r(i) = c\})$
        $w_{c,t+1} \leftarrow proj_{\mathcal{W}}\{w_{c,t} - \eta g_t\}$
    end for
    Return $\{\omega_{c,T}\}_{c \in \mathrm{rg}(\mathcal{C}_r)}$
end for
```

Definition A.1 (TrMean$_\beta$)**.** For $\beta \in [0, \frac{1}{2})$, and a set of vectors $x^j \in \mathbb{R}^d$, $j \in [J]$, their trimmed mean $g = \mathrm{TrMean}_\beta(\{x^1, x^2, \ldots, x^J\})$ is a vector $g \in \mathbb{R}^d$, with each coordinate $g_k = \frac{1}{(1-2\beta)J} \sum_{x \in U_k} x$, for each $k \in [d]$, where U_k is a subset of $\{x_k^1, x_k^2, \ldots, x_k^J\}$ obtained by removing the smallest and largest β fraction of its elements.

Algorithm 3 RECLUSTER()

```
Input: Cluster models $\{\omega_{c,T}\}_{c \in \mathrm{rg}(\mathcal{C}_r)}$, User models $\{w_i\}_{i=1}^m$, Clustering $\mathcal{C}_r$
Output: Improved Clustering $\mathcal{C}'_r$
for all nodes $i \in [m]$ do
    $\mathcal{C}'_r(i) \leftarrow \mathrm{argmin}_{c \in \mathrm{rg}(\mathcal{C}_r)} \mathsf{dist}(w_i, \omega_{c,T})$
end for
return Clustering $\mathcal{C}'_r$.
```

B. Additional Theoretical Guarantees

In this section, we provide additional theoretical results omitted from Section 4. First, we show an example where ℓ_2 norm comes naturally as the $\mathsf{dist}(.,.)$ function, which is the case for our theoretical results.

Proposition B.1. *Suppose that there are m clients, each with a local model $w_i^\star \in \mathbb{R}^d$ and its datapoint $(x, y_i) \in \mathbb{R}^d \times \mathbb{R}$ is generated according to $y_i = \langle w_i^\star, x \rangle + \epsilon_i$. If $x \sim \mathcal{N}(0, I_d)$ and $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2)$, then $KL(p(x,y_i)||p(x,y_j)) = \mathbb{E}_x[KL(p(y_i|x)||p(y_j|x))] = \frac{d}{2\sigma^2}\|w_i - w_j\|^2$.*

Algorithm 4 MERGE()

```
**Input:** Cluster iterates $\{\omega_{c,T}\}_{c\in \mathrm{rg}(\mathcal{C}_r)}$ , Clustering $\mathcal{C}'_r$, Threshold $\lambda$, Size parameter $t$
**Output:** Merged Clustering $\mathcal{C}_{r+1}$, Cluster iterates $\{\omega_{c,T}\}_{c\in \mathrm{rg}(\mathcal{C}_{r+1})}$
$G \leftarrow$ Graph with vertex set $\mathrm{rg}(\mathcal{C}'_r)$ and no edges
for all pairs of clusters $c,c' \in \mathrm{rg}(\mathcal{C}'_r), c \neq c'$ do
    Add edge $(c,c')$ to the graph $G$ if $\mathsf{dist}(w_c, w_{c'}) \leq \lambda$
end for
$\mathcal{C}_{temp} \leftarrow$ Clusters from graph $G$ of size $\geq t$ by correlation clustering of (Bansal et al., 2002).
For each cluster in $\mathcal{C}_{temp}$, merge the nodes of its component clusters to get $\mathcal{C}_{r+1}$
for $c \in \mathrm{rg}(\mathcal{C}_{temp})$ do
    $G_c \leftarrow \{c' \in \mathrm{rg}(\mathcal{C}'_r)$ which merged into $c\}$
    $\omega_{c,T} \leftarrow \frac{1}{|G_c|}\sum_{c'\in G_c} \omega_{c',T}$
end for
return $\mathcal{C}_{r+1}, \{\omega_{c,T}\}_{c\in \mathrm{rg}(\mathcal{C}_{r+1})}$.
```

Hence, we see that minimizing a natural measure (KL divergence) between the conditional distributions $y|x$ for different clients is equivalent to minimizing the ℓ_2 distance of the underlying local models. Note that the above example only serves as a motivation, and our theoretical results hold for a strictly larger class of functions, as defined by our assumptions.

Misclustering Error

Remark B.2 (Improved separation compared with IFCA). Let us now compare the separation with that of IFCA. Note that for IFCA, $\epsilon_1 = 0$, and the separation is $\tilde{\mathcal{O}}(\max\{\frac{\alpha^{-2/5}}{n^{1/5}}, \frac{\alpha^{-1/3}}{n^{1/3}m^{1/6}}\})$, where $\alpha > 0$ is the initialization factor. In the regime where $\alpha = \mathcal{O}(1)$, IFCA requires a separation of $\tilde{\mathcal{O}}(\frac{1}{n^{1/5}})$, which is much worse compared to SR−FCA which requires a separation of $\tilde{\mathcal{O}}(\frac{1}{n})$.

We provide the full restatement of progress made in one REFINE step.

Theorem B.3 (Restatement of Theorem 4.5)**.** *Let $\beta t = \Theta(c_{\min})$, and REFINE() is run with TrimmedMeanGD(β). Provided $\min\{\frac{n^{2/3}\Delta'^{4/3}}{D^{2/3}}, \frac{n^2\Delta'^2}{\hat{L}^2\log(c_{\min})}\} \gtrsim d$, with $0 < \beta < \frac{1}{2}$, where $\Delta' = \Delta - \frac{\mu B}{2} > 0$ and $B = \sqrt{2\hat{L}\epsilon_1/\mu}$. Then, for any constant $\gamma_1 \in (1,2)$ and $\gamma_2 \in (1, 2 - \frac{\mu B}{2\Delta})$, such that after running 1 step of REFINE() with $\eta \leq \frac{1}{L}$, we have*

$$\Pr[C_1 \neq C^\star] \leq \frac{m}{c_{\min}}\exp(-a_1 c_{\min}) + \frac{m}{t}\exp(-a_2 m) + (1-\beta)m(\frac{p}{m})^{\gamma_1} + m(\frac{p}{m})^{\gamma_2} + 8d\frac{m}{t}\exp(-a_3 n\frac{\Delta'}{2\hat{L}}),$$

where $c_{\min}$ is the minimum size of the cluster. Further for some small constants $\rho_1 > 0, \rho_2 \in (0,1)$, we can select β, γ_1 and γ_2 such that for large m,n and Δ', with $B << \frac{2\Delta'}{\mu}$, we have $\Pr[C_1 \neq C^\star] \leq \frac{\rho_1}{m^{1-\rho_2}}p$.

Using single step of REFINE, we obtain the improvement in misclustering error after R steps of REFINE.

Theorem B.4. *[Multi-step REFINE()] If we run R steps of REFINE(), resampling n_i points from $\mathcal{D}_i$ and recompute w_i as in ONE_SHOT for every step of REFINE(), then the probability of error for SR−FCA with R steps of REFINE() is $\Pr[C_R \neq C^\star] \leq \left(\frac{\rho_2}{m^{(1-\rho_1)}}p\right)^R$.*

Remark B.5 (Re-sampling). Note that although the theoretical convergence of Multi-step REFINE() requires resampling of data points in each iteration of REFINE(), we experimentally validate (see Section 5, that this is not required at all.

Remark B.6. In experiments (Section 5), we observe that it is often sufficient to run $1-2$ steps of REFINE(). Since each step of REFINE() reduces the probability of misclusteing by (almost) a factor of $1/m$, very few steps of REFINE() is often sufficient.

Note that the proofs for Theorem 4.4 and Theorem B.3 are provided in Appendix F and Appendix G respectively.

Convergence of cluster iterates:

Remark B.7 (Comparison with IFCA in statistical error). Note that for IFCA, $\epsilon_1 = 0$ and the statistical error rate of IFCA is $\tilde{\mathcal{O}}(1/n)$ (see Theorem 2 in (Ghosh et al., 2022)). Looking at Theorem 4.6, we see that under similar condition ($\epsilon_1 = 0$ and hence $B = 0$), SR−FCA obtains an error rate of $\tilde{\mathcal{O}}(1/\sqrt{n})$, which is weaker than IFCA. This can be thought of the price of initialization. In fact for IFCA, a *good* initialization implies that only a very few clients will be mis-clustered, which was crucially required to obtain the $\tilde{\mathcal{O}}(1/n)$ rate. But, for SR−FCA, we do not have such guarantees which results in a weaker statistical error.

Table 2: Test Accuracy and standard deviations across 5 random seeds on simulated datasets. The highest accuracy is **bold**. SR-FCA is competitve with IFCA and beats it for Rotated CIFAR10.

Baseline	MNIST (inverted)	MNIST (rotated)	CIFAR (rotated)
SR-FCA	**92.03 ±0.30**	**91.66 ± 0.13**	**91.38 ± 0.27**
Local	76.52 ±0.54	85.55 ± 0.19	75.87± 0.33
Global	88.61 ± 0.77	80.88 ±1.55	88.75± 0.52
CFL (Sattler et al., 2019a)	88.30 ± 1.12	80.47 ±0.44	87.59 ± 0.42
LOCAL-KMeans (Ghosh et al., 2019)	10.56 ± 1.31	10.35 ± 0.71	10.00 ±0.20
IFCA (Ghosh et al., 2022)	91.55± 0.81	**91.80 ± 0.25**	86.05 ± 0.43

Remark B.8 (Potential improvement, matching error of IFCA). Our iterates are obtained via TrimmedMeanGD() which assumes β fraction of clients inside each cluster are corrupted. Instead, if we run any federated optimization algorithm which can accommodate low heterogeneity, for instance FedProx (Sahu et al., 2018b), inside each cluster C_R, then we can shave off the Λ term from Theorem 4.6, to obtain convergence to a neighborhood of radius $2B$ of $\omega_c^\star$ for each cluster $c \in C^\star$. We keep this as a future work.

Note that the proof of Theorem 4.6 is provided in Appendix H.

C. Additional Experiments

We provide a detailed description of our experimental setup.

Simulated Datasets: We generate clustered FL datasets from MNIST (LeCun & Cortes, 2010) and CIFAR10 (Krizhevsky et al.) by splitting them into disjoint sets, one per client. For MNIST, by inverting pixel value, we create 2 clusters (referred to as inverted in Table 2) and by rotating the image by 90,180,270 degrees we get 4 clusters. We set $m=100, n=600$. For CIFAR10, we create 2 clusters by rotating the images by 180 degrees and set $m=32, n=3125$. To emulate practical FL scenarios, we assume that only a fraction of the nodes participate in the learning procedure. For Rotated and Inverted MNIST, we assume that all the nodes participate, while for Rotated CIFAR10, 50% of the nodes participate. For MNIST, we train a 2-layer feedforward NN, while for CIFAR10, we train a ResNet9 (Page, 2019). We train Rotated MNIST, Inverted MNIST and Rotated CIFAR10 for 250, 280 and 2400 iterations respectively with 2 refine steps for SR-FCA.

Real Datasets: We sample $m = 50$ machines from FEMNIST and Shakespeare. FEMNIST is a Federated version of EMNIST with data on each client being handwritten symbols from a different person. Shakespeare is a NLP dataset where the task is next character prediction. For FEMNIST, train a CNN for while for Shakespeare we train a 2-layer stacked LSTM. For clustered FL baselines, we tune K, the number of clusters, with $K \in \{2,3,4,5\}$ for FEMNIST and $K \in \{1,2,3,4\}$. We run FEMNIST and Shakespeare for 1000 and 2400 iterations respectively and set number of refine steps to be 1 for SR-FCA.

Test Metrics : The test performance of any baseline is obtained by averaging over the clients, the test performance of each client on its model trained by the baseline. For the local baseline, it is the client's local model and for the global baseline it is the single global model. For SR-FCA and clustered FL baselines, it is the cluster model for the client. Note that we do not present convergence plots as different algorithms run in different number of stages. For simulated datasets, the true clustering $\mathcal{C}^\star$ is known, therefore we report both the test accuracy and misclustering error in Table 2 and Table 3 respectively.

Note that for simulated datasets, we do not compare with the variants of IFCA (FedSoft and ONE_SHOT-IFCA).

Note that the total time to run all experiments including hyperparameter tuning on a single NVIDIA-GeForce-RTX-3090 is 2 weeks.

C.1. Results on Simulated Datasets

Across all datasets, we find that SR-FCA is competitive with or outperforms all other algorithms in terms of both misclustering error and test accuracy.

Comparison with CFL and Local-KMeans: CFL and Local-KMeans use the cosine distance between gradients and l_2

Table 3: Average Misclustering error of clustered FL algorithms on test set across 5 random seeds for simulated datasets. The lowest error is **bold**. SR-FCA is competitive with IFCA for MNIST and beats it for Rotated CIFAR10.

BASELINE	MNIST (INVERTED)	MNIST (ROTATED)	CIFAR (ROTATED)
SR-FCA	**0.0**	**0.0**	**0.0**
CFL (SATTLER ET AL., 2019A)	0.08	0.14	0.18
LOCAL-KMEANS (GHOSH ET AL., 2019)	0.36	0.28	0.38
IFCA (GHOSH ET AL., 2022)	**0.0**	**0.0**	0.50

distance between model weights which are not suitable for NN models. Local-KMeans performs the worst with $\approx 10\%$ test accuracy for simulated datasets. SR-FCA and IFCA use cross-cluster loss and client loss respectively, which are better suited to NN models, thus outperforming these baselines (see Tables 1 and 2).

Comparison with IFCA: On **simulated datasets** (Tables 2 and 3), we find that IFCA recovers $\mathcal{C}^\star$ and outperforms SR-FCA marginally for MNIST datasets. This is due to MNIST being a simpler and easier to learn dataset, even after adding heterogeneity via rotations or inversions. In contrast, for CIFAR10 the learning task is much more difficult, and IFCA, without proper initialization, ends up with all clients in only a single cluster after a few rounds resulting in a misclustering of 0.5, as seen in Table 3. Thus it performs slightly worse than the global baseline in terms of test accuracy, as seen in Table 2. From Table 3, we see that SR-FCA correctly identifies $\mathcal{C}^\star$ and comprehensively beats IFCA in terms of test accuracy.

D. Computational and Communication Complexity

Note that the complexity of the REFINE step is the same as that of IFCA in terms of both computation time and communication since in each case, we need to find the loss of every cluster model on every client's data. The main blowup of $\mathcal{O}(m^2)$ is incurred during ONE_SHOT, which is unavoidable if an initial clustering is not known. We use the comparison of KMeans (Lloyd, 1982) v/s DBSCAN (Ester et al., 1996) or Ward's algorithm where without the initial clustering, we need to perform all pairwise comparisons to check which clients can be clustered together.

In the next section, we will provide theoretical justification for several of our claims and establish the probability of clustering error and convergence rates for the cluster models obtained by TrimmedMeanGD.

E. Proof of Proposition B.1

According to the proposition, for two users i and j, the data is generated by first sampling each coordinate of $x \in \mathbb{R}^d$ from $\mathcal{N}(0,1)$ iid and then computing y as –

$$y_i = \langle x, w_i^\star \rangle + \epsilon_i$$

where $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$. Then, the distribution of $y_i|x$ is $\mathcal{N}(\langle x, w_i^\star\rangle, \sigma^2)$. Therefore, the KL divergence between $y_i|x$ and $y_j|x$ is given by

$$KL(p(y_i|x)||p(y_j|x)) = \frac{\langle w_i^\star - w_j^\star, x\rangle^2}{2\sigma^2}$$

Therefore, if we take expectation wrt x, we have

$$\mathbb{E}_x[KL(p(y_i|x)||p(y_j|x))] = \frac{d\|w_i^\star - w_j^\star\|^2}{2\sigma^2}$$

F. Proof of Theorem 4.4

In ONE_SHOT(), $\mathcal{C}_0 = \mathcal{C}^\star$, if all the edges formed in the graph are correct. This means that if i,j are in the same cluster in $\mathcal{C}^\star$, then $\|w_{i,T} - w_{j,T}\| \le \lambda$ and if i,j are in different clusters, $\|w_{i,T} - w_{j,T}\| > \lambda$.

Note that,

$$w_{i,T} - w_{j,T} = (w_i^\star - w_j^\star) + (w_{i,T} - w_i^\star) - (w_{j,T} - w_j^\star)$$

Now, if we apply triangle inequality, we obtain

$$\mathsf{dist}(w_{i,T}, w_{j,T}) \geq \mathsf{dist}(w_i^\star, w_j^\star) - \Xi_{i,j}, \quad \mathsf{dist}(w_{i,T}, w_{j,T}) \leq \mathsf{dist}(w_i^\star, w_j^\star) + \Xi_{i,j}$$

where $\Xi_{i,j} = \sum_{k=i,j} \mathsf{dist}(w_{k,T}, w_k^\star)$. This decomposition forms the key motivation for our algorithm.

Therefore, if i,j are in the same cluster, then a sufficient condition for edge (i,j) to be incorrect is

$$\lambda \leq \mathsf{dist}(w_i^\star, w_j^\star) + \Xi_{i,j} \implies \Xi_{i,j} \geq \lambda - \epsilon_1$$

Similarly, if i,j are in different clusters, then a sufficient condition for edge (i,j) to be incorrect is

$$\lambda \geq \mathsf{dist}(w_i^\star, w_j^\star) - \Xi_{i,j} \implies \Xi_{i,j} \geq \epsilon_2 - \lambda$$

Therefore, we can set $\Delta_\lambda = \min\{\epsilon_2 - \lambda, \lambda - \epsilon_1\}$, and then a sufficient condition for any edge to be incorrect is $\max_{i,j} \Xi_{i,j} \geq \Delta_\lambda$.

Thus,

$$\begin{aligned}
\Pr[\mathcal{C}^\star \neq \mathcal{C}_0] &\leq \Pr[\text{at least 1 edge is incorrect}] \\
&\leq \Pr[\max_{i,j} \Xi_{i,j} \geq \Delta_\lambda] \\
&\leq \Pr[\max_{i,j} \sum_{k=i,j} \|w_{k,T} - w_k^\star\| \geq \Delta_\lambda] \\
&\leq \Pr[\max_{i,j} \max_{k=i,j} (\|w_{k,T} - w_k^\star\| \geq \frac{\Delta_\lambda}{2}] \\
&\leq \Pr[\max_{i \in [m]} \|w_{i,T} - w_i^\star\| \geq \frac{\Delta_\lambda}{2}]
\end{aligned} \tag{2}$$

The second and third inequalities are obtained by expanding the terms. The fourth inequality is obtained by $\Pr[a+b \geq c] \leq \Pr[\max\{a,b\} \geq c/2]$. For the fifth inequality, we merge $\max_{i,j} \max_{k=i,j}$ into $\max_{i \in [m]}$. As we can see in Equation (2), we need to bound $\|w_{i,T} - w_i^\star\|$ for each node i. The subsequent Lemma allow us to bound this quantities.

Lemma F.1 (Convergence of $w_{i,T}$). *Let $\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}} \lesssim b_1 d$, for some constant $b_1 > 0$. Then, after running* `ONE_SHOT()` *with $\eta \leq \frac{1}{L}$, for some constant $b_2 > 0$, under Assumptions 4.1-4.3, we have*

$$\Pr[\|w_{i,T} - w_i^\star\| \geq \frac{\epsilon_2 - \epsilon_1}{4}] \leq d \ \exp(-n \frac{b_2 \Delta}{\hat{L}\sqrt{d}}),$$

where $\Delta = \frac{\mu}{2}(\frac{\Delta_\lambda}{2} - (1 - \frac{\mu}{L})^{T/2} D)$ and $n = \min_{i \in [m]} n_i$.

This lemma follows from (Yin et al., 2018). The complete proof of this Lemma is present in Appendix F.1.

Now, we can apply Lemma F.1 in Eq (2).

$$\begin{aligned}
\Pr[\mathcal{C}_0 \neq \mathcal{C}^\star] &\leq \Pr[\max_{i \in [m]} \|w_{i,T} - w_i^\star\| \geq \frac{\Delta_\lambda}{2}] \\
&\leq m \max_{i \in [m]} \Pr[\|w_{i,T} - w_i^\star\| \geq \frac{\Delta_\lambda}{2}] \\
&\leq m d \ \exp(-n \frac{b_2 \Delta}{\hat{L}\sqrt{d}})
\end{aligned}$$

For the second inequality, we use $\Pr[\max_{i \in [m]} a_i \geq c] \leq \sum_{i \in [m]} \Pr[a_i \geq c] \leq m \max_{i \in [m]} \Pr[a_i \geq c]$, which follows from union bound.

Note that for $p < 1$, we need the separation to be order of $\Theta(\sqrt{\frac{\log m}{n}})$.

F.1. Proof of Lemma F.1

We utilize results from (Yin et al., 2018), which hold for `TrimmedMeanGD` to analyze convergence for a single node as they yield stronger guarantees under the given assumptions.

Lemma F.2 (Convergence of $w_{i,T}$). *If Assumptions 4.1-4.3 hold, and $\eta \leq \frac{1}{L}$, then*

$$\|w_{i,T} - w_i^\star\| \leq (1-\kappa^{-1})^{T/2} D + \frac{2}{\mu}\Lambda_i \quad \forall i \in [m] \tag{3}$$

where $\kappa = \frac{L}{\mu}$ and Λ_i is a positive random variable with

$$\Pr[\Lambda_i \geq \sqrt{2d}r + 2\sqrt{2}\delta\hat{L}] \leq 2d(1+\frac{D}{\delta})^d \exp(-n\min\{\frac{r}{2\hat{L}}, \frac{r^2}{2\hat{L}^2}\}) \tag{4}$$

for some $r, \delta > 0$.

We provide the proof of this lemma in Appendix G.8.

Using the above Lemma, we can bound the probability $\Pr[\|w_{i,T} - w_i^\star\| \geq \frac{\Delta_\lambda}{2}]$

$$\begin{aligned}
\Pr[\|w_{i,T} - w_i^\star\| \geq \frac{\Delta_\lambda}{2}] &\leq \Pr[2(1-\kappa^{-1})^{T/2} D + \frac{2}{\mu}\Lambda_i + \geq \frac{\Delta_\lambda}{2}] \\
&\leq \Pr[\Lambda_i \geq \Delta], \quad \text{where } \Delta = \frac{\mu}{2}(\frac{\Delta_\lambda}{2} - (1-\kappa^{-1})^{T/2} D) \\
&\leq \Pr[\sqrt{2d}r + 2\sqrt{2}\delta\hat{L} \geq \Delta] \\
&\leq d\exp(-nb_2 \frac{\Delta}{\hat{L}\sqrt{d}})
\end{aligned}$$

for some constants $b_1, b_2, b_3, b_4 > 0$, where we set $r = b_3\hat{L}\max\{\frac{\Delta}{\hat{L}\sqrt{d}}, \sqrt{\frac{\Delta}{\hat{L}\sqrt{d}}}\}$ and $\delta = b_4\frac{\Delta}{\hat{L}}$, and for $b_1 d \leq \frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{4/3}}$, such that $\sqrt{2d}r + 2\sqrt{2}\delta\hat{L} \geq \Delta$ and $n\min\{\frac{r}{2\hat{L}}, \frac{r^2}{2\hat{L}^2}\} > \frac{Dd}{\delta}$ in Lemma F.2.

G. Proof of Theorem B.3

G.1. Preliminaries

First, we define certain random variables and their respective probabilities which we will use throughout this proof. Since the edge based analysis and corresponding clique identification involves a lot of dependent events, we try to decompose the absence/presence of edge into a combination of independent events.

Define,

$$X_{ij} = \begin{cases} 1 & \text{If the edge } (i,j) \text{ in } \mathcal{C}_0 \text{ is incorrect in } \mathcal{C}^\star \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

An edge (i,j) in $\mathcal{C}_0$ is incorrect in $\mathcal{C}^\star$ if either it is present in $\mathcal{C}^\star$ and absent in $\mathcal{C}_0$ or vice versa. We analyze the probability of this event for the case when $\mathcal{C}^\star$ contains the edge (i,j). The case when $\mathcal{C}^\star$ doesn't contain edge (i,j) and it is present in $\mathcal{C}_0$ has exaclty same probability. When $\left\|w_i^\star - w_j^\star\right\| \leq \epsilon_1$, then edge is present is $\mathcal{C}^\star$. If it is absent in $\mathcal{C}_0$, then

$$\begin{aligned}
\Pr[X_{ij} = 1] &\leq \Pr[\Xi_{i,j} \geq \Delta_\lambda] \\
&\leq \Pr[\Lambda_i + \Lambda_j \geq 2\Delta]
\end{aligned}$$

The analysis is similar to the proof of `ONE_SHOT()` in Appendix F.

Note that the random variables $\{X_{ij}\}$ are not independent. We now define independent random variables X_i such that

$$X_i = \begin{cases} 1 & \text{If } \Lambda_i \geq \Delta \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

Thus, we can see that $X_{ij} \leq X_i + X_j$. Additionally,

$$\Pr[X_i = 1] \leq \Pr[\Lambda_i \geq \Delta] \leq \frac{p}{m} \tag{7}$$

This follows from analysis of `ONE_SHOT()` in Appendix F.

We can further generalize this notion to the random variables defined as $Y_{i,\gamma}$.

$$Y_{i,\gamma} = \begin{cases} 1 & \text{If } \Lambda_i \geq \gamma\Delta, \gamma \in (0,2) \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

Then,

$$\Pr[Y_{i,\gamma} = 1] \leq \Pr[\Lambda_i \geq \gamma\Delta] \leq d\exp(-nb_2 \frac{\gamma\Delta}{\hat{L}\sqrt{d}}) = (\frac{p}{m})^{\gamma}$$

Note that the set of random variables $\{Y_{i,\gamma}\}_{i=1}^m$ are mutually independent random variables.

Further, we define the $\omega_c^\star$ for every cluster $c \in \mathrm{rg}(\mathcal{C}_0)$. Let $c' \in \mathcal{C}^\star$ be the cluster label of node c. If $G_c = \{i : i \in [m], \mathcal{C}^\star(i) = c'\}$, which is the set of nodes in c which were from c' in the original clustering, then we can define $\omega_c^\star$ and $F_c(w)$ as

$$\omega_c^\star = \operatorname*{argmin}_{w \in \mathcal{W}} \mathbb{E}[\frac{1}{|G_{c'}|} \sum_{i \in G_{c'}} f_i(w)] \tag{9}$$

$$= \operatorname*{argmin}_{w \in \mathcal{W}} \frac{1}{|G_{c'}|} \sum_{i \in G_{c'}} F_i(w) = \operatorname*{argmin}_{w \in \mathcal{W}} F_c(w) \tag{10}$$

We use this definition of $\omega_c^\star$ in the Appendices G.5 and G.6.

G.2. Analysis of REFINE()

Our goal is to compute total probability of error for `REFINE()` to fail. If we define this error as $\mathcal{C}_1 \neq \mathcal{C}^\star$, then we can define the main sources of error for this event.

1. **$\exists c \in \mathrm{rg}(\mathcal{C}^\star)$ such that no cluster in $\mathcal{C}_0$ has cluster label c :** If the a cluster $c \in \mathrm{rg}(\mathcal{C}^\star)$ is absent in $\mathcal{C}_0$, then subsequent steps of `REFINE()` will never be able to recover it, as they only involve node reclustering and merging existing clusters. The lemma presented below gives an upper bound on the probability of this event.

   Lemma G.1. *Under the conditions of Theorem 4.4 and if $t = \Theta(c_{\min})$, then there exists constant $a_1 > 0$ such that*

   $$\Pr[\exists c \in \mathrm{rg}(\mathcal{C}^\star) \textit{ such that no cluster in } \mathcal{C}_0 \textit{ has cluster label } c] \leq \frac{m}{c_{\min}} \exp(-a_1 c_{\min})$$

   The proof of this Lemma is presented in Appendix G.3

2. **Each cluster $c \in \mathrm{rg}(C)_0$ should have $< \alpha$ fraction of impurities for some $\frac{1}{2} > \beta > \alpha$:** If some cluster has more than α-fraction of impure nodes, then we cannot expect convergence guarantees for `TrimmedMeanGD`$_\beta$.

   The below lemma bounds the probability of this error as

   Lemma G.2. . *For some constants $0 < \alpha < \beta < \frac{1}{2}, a_2 \geq 0, \gamma_1 \in (1,2)$ and $\alpha t = \Theta(m)$, under the conditions in Theorem 4.4, we have*

   $$\Pr[\exists c \in \mathrm{rg}(\mathcal{C}_0) \textit{ which has} > \alpha \textit{ fraction of impurities}] \leq \frac{m}{t} \exp(-a_2 m) + (1-\alpha) m (\frac{p}{m})^{\gamma_1}$$

   The proof of this Lemma is presented in Appendix G.4.

3. **`MERGE()` error:** We define this as the error for the `MERGE()` to fail. Even though `MERGE()` operates after `RECLUSTER()`, `RECLUSTER()` does not change the cluster iterates. The goal of `MERGE()` is to ensure that all clusters in $\mathcal{C}_0$ with the same cluster labels are merged. Therefore, we define `MERGE()` error as the event when either two clusters with same cluster label are not merged or two clusters with different cluster labels are merged. The below lemma bounds this probability.

   Lemma G.3. *If* $\min\{\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}}, \frac{n^2\Delta'^2}{\hat{L}^2\log(c_{\min})}\} \geq u_1 d$ *for some constants* $u_1 > 0$*, then for some constant* $a_3' > 0$*, where* $\Delta' = \Delta - \frac{\mu B}{2} > 0$*, where* $B = \sqrt{\frac{2\hat{L}\epsilon_1}{\mu}}$*, we have*

$$\Pr[\textit{MERGE() Error}] \leq \frac{4dm}{t}\exp(-a_3' n \frac{\Delta'}{2\hat{L}})$$

   The proof of this Lemma is presented in Appendix G.5.

4. **`RECLUSTER()` error:** This event is defined as a node going to the wrong cluster after both `MERGE()` and `REFINE()` operations. After `MERGE()`, each cluster in $\mathcal{C}_0$ corresponds to a single cluster in $\mathcal{C}_1$. Therefore, we incur an error due to the `RECLUSTER()` operation if any node i does not go to the cluster $c \in \mathcal{C}_1$ which has cluster label $\mathcal{C}^\star(i)$. The below lemma provides an upper bound on the probability of this error.

   Lemma G.4. *If* $\min\{\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}}, \frac{n^2\Delta'^2}{\hat{L}^2\log(c_{\min})}\} \geq u_2 d$ *for some constants* $u_2 > 0$*, then for some constants* $a_3'' > 0$ *and* $\gamma_2 \in (1, 2 - \frac{\mu B}{2\Delta})$*, we have*

$$\Pr[\textit{RECLUSTER() error}] \leq 4d\frac{m}{t}\exp(-a_3'' n \frac{\Delta'}{2\hat{L}}) + m(\frac{p}{m})^{\gamma_2} \tag{11}$$

   The proof of this Lemma is presented in Appendix G.6.

The total probability of error after for a single step of `REFINE()` is the sum of probability of errors for these 4 events by the union bound. Therefore,

$$\begin{aligned}\Pr[\mathcal{C}_1 \neq \mathcal{C}^\star] \leq &\frac{m}{c_{\min}}\exp(-a_1 c_{\min}) + \frac{m}{t}\exp(-a_2 m)\\ &+ (1-\beta)m(\frac{p}{m})^{\gamma_1} + 8d\frac{m}{t}\exp(-a_3 n\frac{\Delta'}{2\hat{L}}) + m(\frac{p}{m})^{\gamma_2}\end{aligned}$$

where we set $a_3 = \min\{a_3', a_3''\}$.

For some small constants $\rho_1 > 0, \rho_2 \in (0,1)$, we can choose $\gamma_1 \in (1,2), \beta \in (0, \frac{1}{2})$ and $\gamma_2 \in (1, 2 - \frac{\mu B}{2\Delta})$ such that $(1-\beta)(\frac{p}{m})^{\gamma_1 - 1} + (\frac{p}{m})^{\gamma_2 - 1} \leq \frac{\rho_1}{2m^{1-\rho_2}}$ and for large enough m, Δ' and n, $\frac{m}{c_{\min}}\exp(-a_1 c_{\min}) + \frac{m}{t}\exp(-a_2 m) + 8d\frac{m}{t}\exp(-a_3 n\frac{\Delta'}{2\hat{L}}) \leq \frac{\rho_1}{2m^{1-\rho_2}}p$. This happens because we have terms of $\exp(-m)$, $\exp(-c_{\min})$ and $\exp(-n\Delta')$, which decrease much faster than $\frac{p}{m}$ which has terms of $\mathcal{O}(m\exp(-n\Delta))$, where Δ and Δ' are of the same order. Therefore, the total probability of error can be bounded by

$$\Pr[\mathcal{C}_1 \neq \mathcal{C}^\star] \leq \frac{\rho_1}{m^{1-\rho_2}}p \tag{12}$$

G.3. Proof of Lemma G.1

$$\begin{aligned}&\Pr[\exists c \in \mathrm{rg}(\mathcal{C}^\star) \text{ such that no cluster in } \mathcal{C}_0 \text{ has cluster label } c]\\ &\leq \sum_{c \in \mathcal{C}^\star}\Pr[\text{No cluster in } \mathcal{C}_0 \text{ has cluster label } c]\end{aligned} \tag{13}$$

Here, we use union bound over the clusters for the second inequality. Now, we analyze the probability that no cluster in $\mathrm{rg}(\mathcal{C}_0)$ has cluster label c for some $c \in \mathrm{rg}(\mathcal{C}^\star)$. Consider a cluster in $\mathrm{rg}(\mathcal{C}_0)$. This cluster has cluster label c if a majority of its nodes are from cluster $c \in \mathrm{rg}(\mathcal{C}^\star)$. Since the size of each cluster in $\mathrm{rg}(\mathcal{C}_0)$ is atleast t and there are C clusters in $\mathrm{rg}(\mathcal{C}^\star)$, if all clusters in $\mathrm{rg}(\mathcal{C}_0)$ have $\leq \frac{t}{C}$ nodes from cluster c, then no cluster will have cluster label c.

Assume that the clique formed by nodes from cluster c has r nodes. Then, every node i in cluster c, must have $S_c - r$ edges absent, which correspond to the edges between a node of the clique and those outside it. Thus, we obtain,

$$\begin{aligned}
\Pr[\text{No cluster in } \mathcal{C}_0 \text{ has cluster label } c] \leq & \Pr[\bigcap_{\mathcal{C}^\star(i)=c}\{\sum_{j\neq i,\mathcal{C}^\star(i)=c} X_{ij} > S_c - \frac{t}{C}\}] \\
\leq & \Pr[\sum\sum_{\mathcal{C}^\star(i)=\mathcal{C}^\star(j)=c} X_{ij} > S_c(S_c - \frac{t}{C})] \\
\leq & \Pr[\sum\sum_{\mathcal{C}^\star(i)=\mathcal{C}^\star(j)=c} (X_i + X_j) > S_c(S_c - \frac{t}{C})] \\
\leq & \Pr[\frac{1}{S_c}\sum_{\mathcal{C}^\star(i)=c} X_i > 1 - \frac{t}{CS_c})] \\
\leq & \exp(-\left(1 - \frac{t}{CS_c} - \frac{p}{m}\right)^2 S_c) \\
\leq & \exp(-a_1 c_{\min})
\end{aligned}$$

In the first step, we require each node i to have $S_c - \frac{t}{C}$ wrong edges. For the second inequality, we remove the intersection and thus, the total number of incorrect edges has to be $S_c(S_c - \frac{t}{C})$, since each node has $S_c - \frac{t}{C}$ incorrect edges. For the third inequality, we use $X_{ij} \leq X_i + X_j$ and collect the terms of X_i for the fourth inequality. In the fifth inequality, we obtain a condition on the sum of independent Bernoulli random variables each with mean $\frac{p}{m}$. Therefore, we can apply Chernoff bound for their sum to obtain the fifth inequality.

A necessary condition for us is $1 - \frac{t}{CS_c} - \frac{p}{m} > 0$ which translates to $t < CS_c(1 - \frac{p}{m})$. If we select $t \leq c_{min} - 1$, this inequality is always satisfied. Note that we want the term $\left(1 - \frac{t}{CS_c} - \frac{p}{m}\right)^2 > a_1$, for some positive constant a_1. If we choose $t = \Theta(m)$, which is possible if $t = \Theta(c_{\min})$ as we assume $c_{\min} = \Theta(m)$, then this is satisfied. We use the lower bound a_1 and $S_c \geq c_{\min}$ to obtain the final inequality. Plugging this in Eq (13), we obtain our result.

G.4. Proof of Lemma G.2

$$\begin{aligned}
& \Pr[\exists c \in \mathrm{rg}(\mathcal{C}_0) \text{ which has } \geq \alpha \text{ fraction of impurities}] \\
\leq & \sum_{c \in \mathrm{rg}(\mathcal{C}_0)} \Pr[\text{cluster } c \text{ has } \geq \alpha \text{ fraction of wrong nodes}]
\end{aligned} \tag{14}$$

We use a simple union bound on clusters in $\mathcal{C}_0$ for the above inequality. Let the set of nodes in the cluster c which are from same cluster of $\mathcal{C}^\star$ as the cluster label of c, i.e., which are not impurities, be R_c. Then let $Q_c = |R_c|$. Let Q'_c denote the number of impurities in cluster c.

$$\begin{aligned}
\Pr[\text{cluster } c \text{ has } \geq \alpha \text{ fraction of wrong nodes}] \leq & \Pr[Q'_c \geq \frac{\alpha}{1-\alpha} Q_c] \\
& \Pr[Q'_c \geq \alpha t]
\end{aligned}$$

We use the fact that $Q_c + Q'_c \geq t$, which is the minimum size of any cluster, for the second inequality.

Now, we analyze the probability of a single node to be incorrect. A node is an impurity in cluster c if it has an edge to each of nodes in R_c.

$$\begin{aligned}
\Pr[\text{Node } i \text{ is an impurity in cluster c}] \leq & \Pr[\min_{j \in R_c} \|w_{i,T} - w_{j,T}\| \leq \lambda] \\
\leq & \Pr[\min_{j \in R_c} (\left\|w_i^\star - w_j^\star\right\| - \Xi_{i,j}) \leq \lambda] \\
\leq & \Pr[\Lambda_i + \max_{j \in R_c} \Lambda_j \geq 2\Delta]
\end{aligned} \tag{15}$$

Now, if $\max_{j\in R_c}\Lambda_j \leq \gamma_1\Delta$, for $\gamma_1 \in (1,2)$, then we need $\Lambda_i \geq (2-\gamma_1)\Delta$ for error.

Using the definition of random variables in Appendix G.1

$$\begin{aligned}\Pr[Q_c' \geq \alpha t] &\leq \Pr[Q_c' \geq \alpha t | \max_{j\in R_c}\Lambda_j \leq \gamma_1\Delta] + \Pr[\max_{j\in R_c}\Lambda_j \geq \gamma_1\Delta]\\ &\leq \Pr[\sum_{i=1}^{m} Y_{i,2-\gamma_1} \geq \alpha t] + \Pr[\max_{j\in R_c}\Lambda_j \geq \gamma_1\Delta]\end{aligned}$$

For the first inequality, we use union bound over the value of $\max_{j\in R_c}\Lambda_j$ and for the second inequality, we need atleast αt impurities, so atleast αt of all $Y_{i,2-\gamma_1}$ should be 1.

We now bound the two terms in the final inequality separately.

For the second term, if $\max_{j\in R_c}\Lambda_j \geq \gamma_1\Delta$.

$$\Pr[\max_{j\in R_c}\Lambda_j \geq \gamma_1\Delta] \leq Q_c \Pr[Y_{j,\gamma_1} = 1] \leq Q_c(\frac{p}{m})^{\gamma_1}$$

Here, we use union bound over all elements in R_c for the first inequality and the second inequality is plugging in the value of $\Pr[Y_{j,\gamma_1} = 1]$, which we have already computed.

Now, we need to provide a bound on Q_c. Note that if Q_c denotes the correct number of nodes, which corresponds to the majority of nodes, then $Q_c \leq (1-\alpha)S_c$, where S_c is the size of the cluster c.

For the first term, we can use Chernoff bound as $Y_{i,2-\gamma_1}$ are independent random variables with expectation $\frac{p}{m}$

$$\Pr[\frac{1}{m}\sum_{i=1}^{m} Y_{i,2-\gamma_1} \geq \alpha\frac{t}{m}] \leq \exp(-(\alpha\frac{t}{m} - \mathbb{E}[Y_{i,2-\gamma_1}])^2 m) \leq \exp(-a_2 m)$$

We need $\alpha\frac{t}{m} \geq \mathbb{E}[Y_{i,2-\gamma_1}]$,which implies $\alpha t \geq 1$, since $Y_{i,2-\gamma_1}$ is a bernoulli random variable. Further, we require $\alpha t = \Theta(m)$, so that we can bound the probability using a constant $a_2 \geq 0$. If we choose γ_1 as a constant independent of m, then we are done.

Now, plugging all these inequalities into Eq (14), we get

$$\begin{aligned}&\Pr[\exists c \in \mathrm{rg}(\mathcal{C}_0) \text{ which has } \geq \alpha \text{ fraction of wrong nodes}]\\ &\leq \mathrm{rg}(\mathcal{C}_0)\exp(-a_2 m) + \sum_{c\in \mathrm{rg}(\mathcal{C}_0)} (1-\alpha)S_c(\frac{p}{m})^{\gamma_1}\\ &\leq |\mathrm{rg}(\mathcal{C}_0)|\exp(-a_2 m) + (1-\alpha)m(\frac{p}{m})^{\gamma_1}\\ &\leq \frac{m}{t}\exp(-a_2 m) + (1-\alpha)m(\frac{p}{m})^{\gamma_1}\end{aligned}$$

For the second inequality, we use $\sum_{c\in\mathcal{C}_0} S_c = m$ and for the third inequality, we use $|\mathrm{rg}(\mathcal{C}_0)|t \leq m$.

G.5. Proof of Lemma G.3

First, let $i,j \in [m]$ be a node in cluster $c,c' \in \mathrm{rg}(\mathcal{C}_0)$ respectively such that $\mathcal{C}^\star(j)$ and $\mathcal{C}^\star(i)$ are the cluster labels of clusters c and c' respectively. Then, if we repeat our thresholding analysis for `MERGE()` operation, we obtain

$$\begin{aligned}&\mathsf{dist}(w_i^\star, w_j^\star) - \Psi_{c,c'} \leq \mathsf{dist}(\omega_{c,T}, \omega_{c',T}) \leq \mathsf{dist}(w_i^\star, w_j^\star) + \Psi_{c,c'}\\ &\text{where } \Psi_{c,c'} = \mathsf{dist}(\omega_c^\star, w_i^\star) + \mathsf{dist}(\omega_{c'}^\star, w_j^\star) + \sum_{k=c,c'} \mathsf{dist}(w_{k,T}, w_k^\star)\end{aligned}$$

We obtain the above equations by a simple application of triangle inequality. Here, $\omega_c^\star$ is as defined in Appendix G.1.

To analyze the above quantities, we need to bound $\|\omega_c^\star - \omega_{c,T}\|$ and $\left\|\omega_c^\star - w_j^\star\right\|$ for some $j \in G_c$. The following Lemmas provide these bounds.

Lemma G.5 (Convergence of $\omega_{c,T}$)**.** *If Assumptions 4.1-4.3 hold, and $\eta \le \frac{1}{L}$, then*

$$\|\omega_{c,T} - \omega_c^\star\| \le (1-\kappa^{-1})^{T/2} D + \frac{2}{\mu}\Lambda_c \quad \forall c \in \mathrm{rg}(\mathcal{C}_0) \tag{16}$$

where $\kappa = \frac{L}{\mu}$ and Λ_c is a positive random variable with

$$\begin{aligned} &\Pr[\Lambda_c \ge \sqrt{2d}\frac{r+3\beta s}{1-2\beta} + \sqrt{2}\frac{2(1+3\beta)}{1-2\beta}\delta\hat{L}] \\ &\le 2d(1+\frac{D}{\delta})^d(\exp(-(1-\alpha)S_c n \min\{\frac{r}{2\hat{L}}, \frac{r^2}{2\hat{L}^2}\}) \\ &\quad + (1-\alpha)S_c \exp(-n\min\{\frac{s}{2\hat{L}}, \frac{s^2}{2\hat{L}^2}\})) \end{aligned} \tag{17}$$

for some $r, s, \delta > 0$ where S_c is the size of cluster c.

Proof is presented in Appendix G.7

Lemma G.6 (Distance between cluster minima and node minima)**.** *If Assumptions 4.1-4.3 holds, for all $j \in [m]$, where j is a node in cluster $c \in \mathcal{C}_0$ where $\mathcal{C}^\star(j)$ is the cluster label of node c, we have*

$$\left\|\omega_c^\star - w_j^\star\right\| \le \sqrt{\frac{2\hat{L}\epsilon_1}{\mu}} := B \tag{18}$$

Proof is presented in Appendix G.9.

Now, that we have our required quantities, we are ready to analyze the probability of error after the merge and reclustering operations.

First, we analyze the probabilty of MERGE() operation. Note that if correct nodes of c and c' were from the same cluster $\mathcal{C}^\star$ then, $\left\|w_i^\star - w_j^\star\right\| \le \epsilon_1, \forall i \in G_c, j \in G_{c'}$. If correct nodes of c' and c were from different clusters in $\mathcal{C}^\star$, then, $\left\|w_i^\star - w_j^\star\right\| \ge \epsilon_2, \forall i \in G_c, j \in G_{c'}$. Therefore, the probability of MERGE() error is upper bounded by

$$\begin{align} \Pr[\texttt{MERGE()}\text{ Error}] &\le \Pr[\text{at least 1 edge is incorrect}] \nonumber \\ &\le \Pr[\max_{c,c'} \Psi_{c,c'} \ge \Delta_\lambda] \nonumber \\ &\le \Pr[\max_{c,c'} \sum_{k=c,c'} \frac{2\Lambda_k}{\mu} \ge \Delta_\lambda - 2(1-\kappa^{-1})^{T/2}D - 2B] \nonumber \\ &\le \max_{c \in \mathrm{rg}(\mathcal{C}_0)} \Pr[\Lambda_c \ge \frac{\mu}{2}(\frac{\Delta_\lambda}{2} - (1-\kappa^{-1})^{T/2}D - B)] \nonumber \\ &\le \max_{c \in \mathrm{rg}(\mathcal{C}_0)} \Pr[\Lambda_c \ge \Delta'] \tag{19} \\ &\le \max_{c \in \mathrm{rg}(\mathcal{C}_0)} 4d\exp(-a_3' n \frac{\Delta'}{2\hat{L}}) \nonumber \\ &\le \sum_{c \in \mathrm{rg}(\mathcal{C}_0)} 4d\exp(-a_3' n \frac{\Delta'}{2\hat{L}}) \le \frac{4dm}{t}\exp(-a_3' n \frac{\Delta'}{2\hat{L}}) \tag{20} \end{align}$$

For the second inequality, we expand all the terms of $\Phi_{c,c'}$. We set $\Delta' = \frac{\mu}{2}(\frac{\Delta_\lambda}{2} - (1-\kappa^{-1})^{T/2}D - B)$. Then, we set $r = \Theta(\hat{L}\max\{\frac{\Delta'}{S_c\sqrt{d}\hat{L}}, \sqrt{\frac{\Delta'}{S_c\sqrt{d}\hat{L}}}\})$, $s = \Theta(\hat{L}\max\{\frac{\Delta'}{S_c\sqrt{d}\hat{L}} + \frac{2\log(S_c)}{n}, \sqrt{\frac{\Delta'}{S_c\sqrt{d}\hat{L}} + \frac{2\log(S_c)}{n}}\})$ and $\delta = \Theta(\frac{Dd^{3/2}\hat{L}}{n\Delta'})$. Now, if $d = \Omega(\min\{\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}}, \frac{n^2\Delta'^2}{\hat{L}^2\log(c_{\min})}\})$, such that $\sqrt{2d}\frac{r+3\beta s}{1-2\beta} + \sqrt{2}\frac{2(1+3\beta)}{1-2\beta}\delta\hat{L} \ge \Delta'$, then there exist some constant $a_3' > 0$ such that the second inequality is satisfied by Lemma G.5. We then use the union bound, followed by $|\mathrm{rg}(\mathcal{C}_0)| \le \frac{m}{t}$.

G.6. Proof of Lemma G.4

We can apply our thresholding analysis to

$\|\omega_{c,T}-w_{i,T}\|$ for $c\in\mathrm{rg}(\mathcal{C}_0)$. First, let j be a node in cluster c such that $\mathcal{C}^\star(j)$ is the cluster label of c.

$$\begin{aligned}&\mathsf{dist}(w_j^\star,w_i^\star)+\Phi_{c,i}\leq\mathsf{dist}(\omega_{c,T},w_{i,T})\leq\mathsf{dist}(w_j^\star,w_i^\star)+\Phi_{c,i}\\&\text{where }\Phi_{c,i}=\mathsf{dist}(\omega_{c,T},\omega_c^\star)+\mathsf{dist}(\omega_c^\star,w_j^\star)+\mathsf{dist}(w_{i,T},w_i^\star)\end{aligned}$$

From Appendix F and Appendix G.5, we have bounds for all the terms involved. Note that after merging, each cluster in $\mathcal{C}^\star$ should have only 1 cluster in $\mathcal{C}_1$. Therefore, after we recluster according to $\|\omega_{c,T}-w_{i,T}\|$, we incur an error if i goes to the wrong cluster. Suppose that the c corresponds to the correct cluster for i and c' is the cluster to which it is assigned , with $c,c'\in\mathrm{rg}(\mathcal{C}_1),c\neq c'$. Then,

$$\begin{aligned}\Pr[\text{Reclustering Error}]\leq&\Pr[\max_{i\in[m]}\max_{c'\neq c}\|\omega_{c',T}-w_{i,T}\|\leq\|\omega_{c,T}-w_{i,T}\|]\\\leq&\Pr[\max_{i\in[m]}\max_{c'\neq c}\epsilon_2-\Phi_{c',i}\leq\epsilon_1+\Phi_{c,i}]\\\leq&\Pr[\max_{i\in[m]}\max_{c'\in\mathcal{C}_0'}\Phi_{c,i}\geq\frac{\epsilon_2-\epsilon_1}{2}]\\\leq&\Pr[\max_{i\in[m]}\max_{c'\in\mathcal{C}_0'}(\Lambda_c+\Lambda_i)\geq\Delta+\Delta'] &(21)\\\leq&\Pr[\max_{c\in\mathcal{C}_0'}\Lambda_c\geq\Delta'-(\gamma_2-1)\Delta]+\Pr[\max_{i\in[m]}\Lambda_i\geq\gamma_2\Delta]\\\leq&\max_{c\in\mathrm{rg}(\mathcal{C}_0)'}\Pr[\Lambda_c\geq\Delta'']+\max_{i\in m}\Pr[\Lambda_i\geq\gamma_2\Delta] &(22)\end{aligned}$$

For the second inequality, we use the thresholding analysis on $\|\omega_{c,T}-w_{i,T}\|$. For the third inequality, we rearrange the terms and combine max over $c'\neq c$ with c, and use. For the fourth inequality, we expand the terms of $\Phi_{c,T}$ and substitute the values of Δ and Δ', using the inequality $\Delta_\lambda\leq\frac{\epsilon_2-\epsilon_1}{2}$. For the fifth inequality, we use consider some $\gamma_2\in(1,2-\frac{\mu B}{2\Delta})$ and break the terms using union bound such that $\Delta''=\Delta'-(\gamma_2-1)\Delta\geq0$. Finally, we use the union bound on $c\in\mathrm{rg}(\mathcal{C}_0)'$ and $i\in[m]$.

Now, we bound the two terms in Eq (22) separately. The second term can be bounded in terms of Y_{i,γ_2}. Thus,

$$\max_{i\in[m]}\Pr[\Lambda_i\geq\gamma_2\Delta]=\max_{i\in[m]}\Pr[Y_{i,\gamma_2}=1]\leq m(\frac{p}{m})^{\gamma_2}\tag{23}$$

We use expectation of Y_{i,γ_2} calculated in Appendix G.4 and then bound max by sum.

For the first term, our analysis is similar to that of `MERGE()` error. Assume that there is some constant $u_2>1$ such that $\Delta''\geq u_2\Delta'$. We set $\delta=\Theta(\frac{Dd^{3/2}\hat{L}}{n\Delta'})$, $r=\Theta(\hat{L}\max\{\frac{\Delta'}{S_c\sqrt{d}\hat{L}},\sqrt{\frac{\Delta'}{S_c\sqrt{d}\hat{L}}}\})$, $s=\Theta(\hat{L}\max\{\frac{\Delta'}{S_c\sqrt{d}\hat{L}}+\frac{2\log(S_c)}{n},\sqrt{\frac{\Delta'}{S_c\sqrt{d}\hat{L}}+\frac{2\log(S_c)}{n}}\})$, and if $d=\Omega(\min\{\frac{n^{2/3}\Delta^{4/3}}{D^{2/3}\hat{L}^{2/3}},\frac{n^2\Delta'^2}{\hat{L}^2\log(c_{\min})}\})$, such that $\sqrt{2d}\frac{r+3\beta s}{1-2\beta}+\sqrt{2}\frac{2(1+3\beta)}{1-2\beta}\delta\hat{L}\geq\Delta'$, then there exist some constant $a_3''>0$ such that the second inequality is satisfied by Lemma G.5. We then use the union bound, followed by $|\mathrm{rg}(\mathcal{C}_0)|\leq\frac{m}{t}$.

$$\begin{aligned}\max_{c\in\mathrm{rg}(\mathcal{C}_0)'}\Pr[\Lambda_c\geq\Delta'']\leq&\max_{c\in\mathrm{rg}(\mathcal{C}_0)'}4d\exp(-a_3''n\frac{\Delta'}{2\hat{L}}) &(24)\\\leq&\sum_{c\in\mathrm{rg}(\mathcal{C}_0)'}4d\exp(-a_3''n\frac{\Delta'}{2\hat{L}}) &(25)\\\leq&\frac{4dm}{t}\exp(-a_3''n\frac{\Delta'}{2\hat{L}}) &(26)\end{aligned}$$

G.7. Proof of Lemma G.5

First, we use an intermediate Lemma from (Yin et al., 2018). This characterizes the behavior of $TrimmedMean_\beta$ gradient estimator.

Lemma G.7 (TrimmedMean Estimator Variance). *Let $g_c(w)$ be the output of TrMean_β estimator for cluster $c \in \mathcal{C}_0$ with size of cluster S_c. If Assumption 4.1,4.2 and 4.3 holds, then*

$$\begin{aligned}
&\|g_c(w)-\nabla F_c(w)\| \leq \Lambda \\
&\textit{where } \Pr[\Lambda \geq \sqrt{2d}\frac{r+3\beta s}{1-2\beta}+\sqrt{2}\frac{2(1+3\beta)}{1-2\beta}\delta\hat{L}] \\
&\leq 2d(1+\frac{D}{\delta})^d\Bigg(\exp(-(1-\alpha)S_c n\min\{\frac{r}{2\hat{L}},\frac{r^2}{2\hat{L}^2}\}) \\
&\quad +(1-\alpha)S_c\exp(-n\min\{\frac{s}{2\hat{L}},\frac{s^2}{2\hat{L}^2}\})\Bigg)
\end{aligned} \tag{27}$$

for some $r,s,\delta > 0$.

Proof. The proof of this Lemma follows from coordinate-wise sub-exponential distribution of ∇F_c. Since loss per sample $f(w,z)$ is Lipschitz in each of its coordinates with Lipschitz constant L_k for $k \in [d]$. Thus, $F_c(w)$ is also L_k-Lipschitz for each coordinate $k \in [d]$ from Corollary I.6. Now, every subgaussian variable with variance σ^2 is σ-sub exponential. Thus, each coordinate of $\nabla_w f(w,z)$ is $\hat{L}$-sub-exponential, since $\hat{L} > L_k, \forall k \in [d]$. The remainder of proof can be found in Appendix E.1 in (Yin et al., 2018). □

Now, using the above Lemma, we can bound the iterate error for a cluster $c \in \mathcal{C}_0$. Consider $\|\omega_{c,t+1}-\omega_c^\star\|^2$,

$$\begin{aligned}
\|\omega_{c,t+1}-\omega_c^\star\| &\leq \|proj_{\mathcal{W}}\{\omega_{c,t}-\eta\nabla g(\omega_{c,t})\}-\omega_c^\star\| \\
&\leq \|\omega_{c,t}-\eta\nabla g(\omega_{c,t})-\omega_c^\star\| \\
&\leq \|\omega_{c,t}-\eta\nabla F(\omega_{c,t})-\omega_c^\star\|+\eta\|g(\omega_{c,t})-\nabla F(\omega_{c,t})\| \\
&\leq \|\omega_{c,t}-\eta\nabla F(\omega_{c,t})-\omega_c^\star\|+\eta\Lambda
\end{aligned}$$

Now, we bound $\|\omega_{c,t}-\eta\nabla F(\omega_{c,t})-\omega_c^\star\|^2$ using μ-strong convexity and L-smoothness of F_c. The analysis is similar to the convergence analysis in Appendix F.1. Thus, for $\eta \leq \frac{1}{L}$

$$\|\omega_{c,t}-\eta\nabla F(\omega_{c,t})-\omega_c^\star\|^2 \leq (1-\eta\mu)\|\omega_{c,t}-\omega_c^\star\|^2$$

Using this bound we can analyze the original term with $\|\omega_{c,t+1}-\omega_c^\star\|$.

$$\begin{aligned}
\|\omega_{c,t+1}-\omega_c^\star\| &\leq \sqrt{1-\eta\mu}\|\omega_{c,t}-\omega_c^\star\|+\eta\Lambda \\
\|\omega_{c,T}-\omega_c^\star\| &\leq (1-\eta\mu)^{T/2}\|\omega_{c,0}-\omega_c^\star\|+\eta\Lambda(\sum_{t=0}^{T-1}(1-\eta\mu)^{t/2}) \\
&\leq (1-\kappa^{-1})^{T/2}\|\omega_{c,0}-\omega_c^\star\|+\eta\Lambda(\sum_{t=0}^{\infty}(1-\frac{\eta\mu}{2})^t) \\
&\leq (1-\kappa^{-1})^{T/2}D+\frac{2}{\mu}\Lambda
\end{aligned}$$

For the second inequality, we use $\kappa = \frac{L}{\mu}$ and unroll the recursion for T steps. For the third inequality, we use $\sqrt{1-x} \leq 1-\frac{x}{2}$ and upper bound the finite geometric sum by its infinite counterpart. Finally we use the boundedness of $\mathcal{W}$ and the sum of the geometric series to get our result.

G.8. Proof of Lemma F.2

We present the proof for this lemma here as it is a corollary of Lemma G.5.

We utilize the intermediate Lemma G.7. Now, if we set $\alpha = \beta = 0$ and $S_c = 1$, we obtain the generalization guarantee for GD on a single node $i \in [m]$. Further, we do not need the terms of s as they appear with β, and thus, we can choose s very large, so that we can ignore its contribution to error probability. The remainder of the proof follows that of Lemma G.5.

G.9. Proof of Lemma G.6

Since F_c is $\hat{L}$-Lipshchitz and μ-strongly convex with minima $\omega_c^\star$,

$$\begin{aligned}
F_c(w_i^\star)-F_c(\omega_c^\star)&=\frac{F_i(w_i^\star)-F_i(\omega_c^\star)}{Q_c}+\sum_{j\neq i,\mathcal{C}_0(j)=c}\frac{F_j(w_i^\star)-F_j(\omega_c^\star)}{Q_c}\\
&\leq\frac{F_i(w_i^\star)-F_i(\omega_c^\star)}{Q_c}+\sum_{j\neq i,\mathcal{C}_0(j)=c}\frac{F_j(w_i^\star)-F_j(w_j^\star)}{Q_c}\\
&\leq-\frac{\mu\|w_i^\star-\omega_c^\star\|^2}{2Q_c}+\sum_{j\neq i,\mathcal{C}_0(j)=c}\frac{\hat{L}\|w_i^\star-w_j^\star\|}{Q_c}\\
\frac{\mu}{2}\|w_i^\star-\omega_c^\star\|^2&\leq-\frac{\mu\|w_i^\star-\omega_c^\star\|^2}{2Q_c}+\frac{(Q_c-1)\hat{L}\epsilon_1}{Q_c}\\
\frac{\mu}{2}\|w_i^\star-\omega_c^\star\|^2&\leq-\frac{\mu\|w_i^\star-\omega_c^\star\|^2}{2Q_c}+\frac{(Q_c-1)\hat{L}\epsilon_1}{Q_c}\\
\|w_i^\star-\omega_c^\star\|^2&\leq\frac{2\hat{L}\epsilon_1}{\mu}\\
\|w_i^\star-\omega_c^\star\|&\leq\sqrt{\frac{2\hat{L}\epsilon_1}{\mu}}
\end{aligned}$$

For the first equation, we expand F_c into its component terms, where Q_c denotes the number of correct nodes in cluster c. For the second inequality, we use the fact that $w_j^\star=\operatorname{argmin}_{w\in\mathcal{W}}F_j(w)$. For the third inequality, we use strong-convexity of F_i and $\hat{L}$-Lipschitzness for $F_j, j\neq i$. For the fourth inequality, we use a lower bound on $F_c(w_i^\star)-F_c(\omega_c^\star)$ using μ-strong convexity of F_c. Finally, we manipulate the remaining terms to obtain the final bound.

H. Proof of Theorem 4.6

By Theorem B.3, $\mathcal{C}_R\neq\mathcal{C}^\star$, with probability $\left(\frac{\rho_2}{m^{(1-\rho_1)}}p\right)^R$. For the $(R+1)^{th}$ step, we bound probability of error by 1. Therefore, with probability $1-\exp(-\frac{5}{8}R)p$. For the $(R+1)^{th}$ step, we optimize the cluster iterates from `TrimmedMeanGD()` to improve convergence instead of clustering error. Since $\mathcal{C}_{R+1}=\mathcal{C}_R$, each cluster in $\mathcal{C}_{R+1}$ maps to some cluster in $\mathcal{C}^\star$. Without loss of generality, assume that cluster $c\in\operatorname{rg}(\mathcal{C}_{R+1})$ maps to the same cluster $c\in\mathcal{C}$. Now, if $\{c_1,c_2,...,c_l\}$ are the clusters in $\mathcal{C}_R$ which merged to form cluster $c\in\operatorname{rg}(\mathcal{C}_{R+1})$. Then, we can write

$$\begin{aligned}
\|\omega_{c,T}-\omega_c^\star\|&=\left\|\frac{1}{l}\sum_{j=1}^{l}(\omega_{c_j,T}-\omega_c^\star)\right\|\\
&\leq\frac{1}{l}\sum_{j=1}^{l}\|\omega_{c_j,T}-\omega_c^\star\|\\
&\leq\frac{1}{l}\sum_{j=1}^{l}(\left\|\omega_{c_j,T}-\omega_{c_j}^\star\right\|+\left\|\omega_{c_j}^\star-\omega_c^\star\right\|)
\end{aligned}$$

For the first inequality, we used the definition of $\omega_{c,T}$ from `MERGE()`. For the second inequality, we used the triangle inequality for the l elements. The third inequality is obtained by using triangle inequality and adding and subtracting $\omega_{c_j}^\star$ as defined in Appendix G.1.

Now, consider the set of nodes $\{i_1,i_2,...,i_l\}\subseteq[m]$, such that $i_j\in c_j\forall j\in[l]$ and $\mathcal{C}^\star(i_j)=c\forall j\in[l]$. Therefore, we can split each term of $\left\|\omega_{c_j}^\star - \omega_c^\star\right\|$ as –

$$\begin{aligned}\|\omega_{c,T}-\omega_c^\star\| &\le \frac{1}{l}\sum_{j=1}^{l}\left(\left\|\omega_{c_j,T}-\omega_{c_j}^\star\right\|+\left\|\omega_{c_j}^\star-w_{i_j}\right\|+\left\|w_{i_j}-\omega_c^\star\right\|\right)\\ &\le \frac{1}{l}\sum_{j=1}^{l}\left\|\omega_{c_j,T}-\omega_{c_j}^\star\right\|+2B\end{aligned}$$

From Lemma G.6, since i_j contributes to both clusters c_j and $c^\star$, we can bound the difference from their minima by B. Further, we can use Lemma G.5 and the Lemma G.7, which is adapted from Theorem 4 in (Yin et al., 2018),to bound the convergence of $\left\|\omega_{c_j,T}-\omega_{c_j}^\star\right\|$. If we set $\delta = \frac{1}{nS_{c_j}\hat{L}D}$ and

$$\begin{aligned}r &= \hat{L}\max\{\frac{8d}{nS_{c_j}}\log(1+nS_c\hat{L}D), \sqrt{\frac{8d}{nS_{c_j}}\log(1+nS_c\hat{L}D)}\}\\ s &= \hat{L}\max\{\frac{4d}{n}(d\log(1+nS_{c_j}\hat{L}D)+\log m), \sqrt{\frac{4d}{n}(d\log(1+nS_{c_j}\hat{L}D)+\log m)}\}\end{aligned}$$

where S_{c_j} is the size of cluster c_j, we obtain

$$\|\omega_{c,T}-\omega_c^\star\| \le (1-\kappa^{-1})^{T/2}D+\Lambda'+2B$$

where

$$\Lambda' = \mathcal{O}\left(\frac{\hat{L}d}{1-2\beta}\left(\frac{\beta}{\sqrt{n}}+\frac{1}{\sqrt{nc_{\min}}}\right)\sqrt{\log(n\max_{j\in[l]}S_{c_j}\hat{L}D)}\right)$$

We can further upper bound $\max_{j\in[l]S_{c_j}}$ by m. Now, the probability of error for each cluster $c \in \mathrm{rg}(\mathcal{C}_R)$ for given values of r and s is $\frac{4d}{(1+nc_{\min}\hat{L}D)^d}$, therefore, we can use union bound and multiply this probability of error by $\mathrm{rg}(\mathcal{C}_R) \le \frac{m}{t}$. Since $t=\Theta(c_{\min})$, we can upper bound this by $\frac{mu''}{c_{\min}}$ for some positive constant $c_{\min}$.

I. Additional Definitions and Lemmas

We start with reviewing the standard definitions of strongly convex and smooth functions $f:\mathbb{R}^d \mapsto \mathbb{R}$.

Definition I.1. f is μ-strongly convex if $\forall w,w'$, $f(w') \ge f(w)+\langle\nabla f(w), w'-w\rangle+\frac{\mu}{2}\|w'-w\|^2$.

Definition I.2. f is L-smooth if $\forall w,w'$, $\|\nabla f(w)-\nabla f(w')\| \le L\|w-w'\|$.

Definition I.3. f is L_k Lipschitz for every coordinate $k\in[d]$ if, $|\partial_k f(w)| \le L_k$, where $\partial_k f(w)$ denotes the k-th coordinate of $\nabla f(w)$.

Lemma I.4. *If $f,g:\mathbb{R}^d \to \mathbb{R}$ are two μ-strongly convex functions on a domain $\mathcal{W}$. Then, $\frac{f+g}{2}$ is also μ-strongly convex on the same domain.*

Proof. If f and g are μ-strongly convex on a domain $\mathcal{W}$, then for any $w_1,w_0\in\mathcal{W}$

$$\begin{aligned}f(w_1) &\ge f(w_0)+\langle\nabla f(w_0), w_1-w_0\rangle+\frac{\mu}{2}\|w_1-w_0\|^2\\ g(w_1) &\ge g(w_0)+\langle\nabla g(w_0), w_1-w_0\rangle+\frac{\mu}{2}\|w_1-w_0\|^2\end{aligned}$$

Adding the above equations, we get

$$\frac{f(w_1)+g(w_1)}{2} \ge \frac{f(w_0)+g(w_0)}{2}+\left\langle\frac{\nabla f(w_0)+\nabla g(w_0)}{2}, w_1-w_0\right\rangle+\frac{\mu}{2}\|w_1-w_0\|^2$$

Thus, $\frac{f+g}{2}$ is also μ-strongly convex. □

Lemma I.5. *If $f,g:\mathbb{R}^d\to\mathbb{R}$ are two L-smooth functions on a domain $\mathcal{W}$. Then, $\frac{f+g}{2}$ is also L-smooth on the same domain.*

Corollary I.6. *If $f,g:\mathbb{R}^d\to\mathbb{R}$ are two L-Lipschitz functions on a domain $\mathcal{W}$. Then, $\frac{f+g}{2}$ is also L-Lipschitz on the same domain.*

Proof. Consider the following term for any $w_1,w_0\in\mathcal{W}$

$$\begin{aligned}&\left\|\frac{\nabla f(w_1)+\nabla g(w_1)}{2}-\frac{\nabla f(w_0)+\nabla g(w_0)}{2}\right\|\\&\leq\frac{1}{2}\|(\nabla f(w_1)-\nabla f(w_0))+(\nabla g(w_1)-\nabla g(w_0))\|\\&\leq\frac{1}{2}(\|\nabla f(w_1)-\nabla f(w_0)\|+\|\nabla g(w_1)-\nabla g(w_0)\|)\\&\leq\frac{1}{2}(L\|w_1-w_0\|+L\|w_1-w_0\|)\\&\leq L\|w_1-w_0\|\end{aligned}$$

In the second inequality, we use the triangle inequality of norms. For the third inequality, we use the L-smoothness of f and g. Thus, $\frac{f+g}{2}$ is also L-smooth The proof of the corollary is same as above, by replacing terms of ∇f and ∇g by f and g respectively. □

Lemma I.7. *If each coordinate of a function $f:\mathbb{R}^d\to\mathbb{R}$ is L_k-Lipschitz for $k\in[d]$ on the domain $\mathcal{W}$, then f is $\hat{L}=\sqrt{\sum_{k=1}^d L_k^2}$-Lipschitz on the same domain $\mathcal{W}$.*

Proof. Consider $w_1,w_0\in\mathcal{W}$.Define a sequence of variables $\{w[k]=((w_1)_1,(w_1)_2...,(w_1)_k,(w_0)_{k+1},...(w_0)_d)^\intercal\}_{k=0}^d$. Then, $w_1=w[d]$ and $w_0=w[0]$

$$\begin{aligned}|f(w_1)-f(w_0)|&=\left|\sum_{k=1}^d(f(w[k])-f(w[k-1]))\right|\\&=\sum_{k=1}^d L_k|(w_1)_k-(w_0)_k|\end{aligned}$$

The second inequality follows by using triangle rule. Then, $f(w[k])$ and $f(w[k-1])$ differ only in the k^{th} coordinate, so we apply L_k coordinate-wise Lipschitzness. Now, consider a random variable $v\in\mathbb{R}^d$ such that $v_k=L_k\frac{|(w_1)_k-(w_0)_k|}{(w_1)_k-(w_0)_k}$ if $(w_1)_k-(w_0)_k\neq 0$, else 0. Then,

$$\begin{aligned}\sum_{k=1}^d L_k|(w_1)_k-(w_0)_k|&=\langle v,w_1-w_0\rangle\\&\leq\|v\|\|w_1-w_0\|\\&\leq\sqrt{\sum_{k=1}^d L_k^2}\|w_1-w_0\|\end{aligned}$$

Here, we use the Cauchy-Schwartz inequality for the second step. Then, note that each coordinate of v is bounded by L_k. □