An Efficient Framework for Clustered Federated Learning

Avishek Ghosh[*]
Dept of EECS, UC Berkeley
Berkeley, CA 94720
avishek_ghosh@berkeley.edu

Jichan Chung[*]
Dept of EECS, UC Berkeley
Berkeley, CA 94720
jichan3751@berkeley.edu

Dong Yin[*]
DeepMind
Mountain View, CA 94043
dongyin@google.com

Kannan Ramchandran
Dept of EECS, UC Berkeley
Berkeley, CA 94720
kannanr@berkeley.edu

Abstract

We address the problem of Federated Learning (FL) where users are distributed and partitioned into clusters. This setup captures settings where different groups of users have their own objectives (learning tasks) but by aggregating their data with others in the same cluster (same learning task), they can leverage the strength in numbers in order to perform more efficient Federated Learning. We propose a new framework dubbed the Iterative Federated Clustering Algorithm (IFCA), which alternately estimates the cluster identities of the users and optimizes model parameters for the user clusters via gradient descent. We analyze the convergence rate of this algorithm first in a linear model with squared loss and then for generic strongly convex and smooth loss functions. We show that in both settings, with good initialization, IFCA converges at an exponential rate, and discuss the optimality of the statistical error rate. In the experiments, we show that our algorithm can succeed even if we relax the requirements on initialization with random initialization and multiple restarts. We also present experimental results showing that our algorithm is efficient in non-convex problems such as neural networks and outperforms the baselines on several clustered FL benchmarks created based on the MNIST and CIFAR-10 datasets by $5 \sim 8\%$.

1 Introduction

In many modern data-intensive applications such as recommendation systems, image recognition, and conversational AI, distributed computing has become a crucial component. In many applications, data are stored in end users' own devices such as mobile phones and personal computers, and in these applications, fully utilizing the on-device machine intelligence is an important direction for next-generation distributed learning. Federated Learning (FL) [27, 16, 26] is a recently proposed distributed computing paradigm that is designed towards this goal, and has received significant attention. Many statistical and computational challenges arise in Federated Learning, due to the highly decentralized system architecture. In this paper, we propose an efficient algorithm that aims to address one of the major challenges in FL—dealing with heterogeneity in the data distribution.

In Federated Learning, since the data source and computing nodes are end users' personal devices, the issue of data heterogeneity, also known as non-i.i.d. data, naturally arises. Exploiting data

[*]Equal contribution

Preprint. Under review.

heterogeneity is particularly crucial in applications such as recommendation systems and personalized advertisement placement, and it benefits both the users' and the enterprises. For example, mobile phone users who read news articles may be interested in different categories of news like politics, sports or fashion; advertisement platforms might need to send different categories of ads to different groups of customers. These indicate that leveraging the heterogeneity among the users is of potential interest—on the one hand, each machine itself may not have enough data and thus we need to better utilize the similarity among the users; on the other hand, if we treat the data from all the users as i.i.d. samples, we may not be able to provide personalized predictions. This problem has recently received much attention [37, 35, 14].

In this paper, we study one of the formulations of FL with non-i.i.d. data, i.e., the *clustered Federated Learning* [35, 25]. We assume that the users are partitioned into different clusters; for example, the clusters may represent groups of users interested in politics, sports, etc, and our goal is to train models for every cluster of users. We note that cluster structure is very common in applications such as recommender systems [34, 22]. The main challenge of our problem is that the *cluster identities of the users are unknown*, and we have to simultaneously solve two problems: identifying the cluster membership of each user and optimizing each of the cluster models in a distributed setting. In order to achieve this goal, we propose a framework and analyze a distributed method, named the *Iterative Federated Clustering Algorithm (IFCA)* for clustered FL. The basic idea of our algorithm is a strategy that alternates between estimating the cluster identities and minimizing the loss functions, and thus can be seen as an Alternating Minimization algorithm in a distributed setting. Here, we emphasize that clustered Federated Learning is not the only approach to modeling the non-i.i.d. nature of the problem, and different algorithms may be more suitable for different application scenarios; see Section 2 for more discussions. That said, our approach to modeling and the resulting IFCA framework is certainly an important and relatively unexplored direction in Federated Learning.

Main contributions: We establish convergence rates of our algorithm, for both linear models and general strongly convex losses under the assumption of good initialization. We prove exponential convergence speed, and for both settings, we can obtain *near optimal* statistical error rates in certain regimes. We also present experimental evidence of its performance in practical settings: We show that our algorithm can succeed even if we relax the requirements on initialization with random initialization and multiple restarts; and we also present results showing that our algorithm is efficient in non-convex problems such as neural networks, and outperforms baseline algorithms on two clustered FL benchmarks created based on the MNIST and CIFAR-10 datasets by $5 \sim 8\%$.

We would also like to mention that our theoretical analysis makes contributions to statistical estimation problems with latent variables in distributed settings. In fact, both mixture of regressions [6] and mixture of classifiers [38] can be considered as special cases of our problem in the centralized setting. We discuss more about these algorithms in Section 2.

Notation: We use $[r]$ to denote the set of integers $\{1, 2, \ldots, r\}$. We use $\|\cdot\|$ to denote the ℓ_2 norm of vectors. We use $x \gtrsim y$ if there exists a sufficiently large constant $c > 0$ such that $x \geq cy$, and define $x \lesssim y$ similarly. We use $\mathrm{poly}(m)$ to denote a polynomial in m with arbitrarily large constant degree.

2 Related work

During the preparation of the initial draft of this paper, we became aware of a concurrent and independent work by Mansour et al. [25], in which the authors propose clustered FL as one of the formulations for personalization in Federated Learning. The algorithms proposed in our paper and by Mansour et al. are similar. However, our paper makes an important contribution by establishing the *convergence rate* of the *population loss function* under good initialization, which simultaneously guarantees both convergence of the training loss and generalization to test data; whereas in [25], the authors provided only *generalization* guarantees. We discuss other related work in the following.

Federated Learning and non-i.i.d. data: Learning with a distributed computing framework has been studied extensively in various settings [49, 31, 21]. As mentioned in Section 1, Federated Learning [27, 26, 16, 12] is one of the modern distributed learning frameworks that aims to better utilize the data and computing power on edge devices. A central problem in FL is that the data

on the users' personal devices are usually non-i.i.d. Several formulations and solutions have been proposed to tackle this problem. A line of research focuses on learning a single global model from non-i.i.d. data [48, 33, 20, 36, 23, 29]. Other lines of research focus more on learning personalized models [37, 35, 7]. In particular, the MOCHA algorithm [37] considers a multi-task learning setting and forms a deterministic optimization problem with the correlation matrix of the users being a regularization term. Our work differs from MOCHA since we consider a statistical setting with cluster structure. Another approach is to formulate Federated Learning with non-i.i.d. data as a meta learning problem [3, 14, 7]. In this setup, the objective is to first obtain a single global model, and then each device fine-tunes the model using its local data. The underlying assumption of this formulation is that the data distributions among different users are similar, and the global model can serve as a good initialization. Here, we would like to mention that this type of approach may not be suitable for applications where the data distributions can form clusters and the distributions of different clusters differ significantly. The formulation of clustered FL has been considered in several recent works [35, 9]. Among them, [35] identifies the cluster structure using cosine similarity, but does not provide theoretical guarantees, and [9] relies on a centralized clustering algorithm, and thus requires high computational cost on the center machine and may not be suitable for large models such as deep neural networks.

Latent variable problems: As mentioned in Section 1, our formulation can be considered as a statistical estimation problem with latent variables in a distributed setting, and the latent variables are the cluster identities. Latent variable problem is a classical topic in statistics and non-convex optimization; examples include Gaussian mixture models (GMM) [43, 19], mixture of linear regressions [6, 42, 47], and phase retrieval [8, 28]. Expectation Maximization (EM) and Alternating Minimization (AM) are two popular approaches to solving these problems. Despite the wide applications, their convergence analyses in the finite sample setting are known to be hard, due to the non-convexity nature of their optimization landscape. In recent years, some progress has been made towards understanding the convergence of EM and AM in the centralized setting [30, 4, 46, 1, 41]. For example, if started from a suitable point, they have fast convergence rate, and occasionally they enjoy super-linear speed of convergence [43, 10]. In this paper, we provide new insights to these algorithms in the FL setting.

3 Problem formulation

We begin with a standard statistical learning setting of empirical risk minimization (ERM). Our goal is to learn parametric models by minimizing some loss functions defined by the data. We consider a distributed learning setting where we have one center machine and m worker machines (i.e., each worker machine corresponds to a user in the Federated Learning framework). The center machine and worker machines can communicate with each other using some predefined communication protocol. We assume that there are k different data distributions, $\mathcal{D}_1, \ldots, \mathcal{D}_k$, and that the m machines are partitioned into k disjoint clusters, $S_1^*, \ldots, S_k^*$. We assume no knowledge of the cluster identity of each machine, i.e., the partition $S_1^*, \ldots, S_k^*$ is not revealed to the learning algorithm. We assume that every worker machine $i \in S_j^*$ contains n i.i.d. data points $z^{i,1}, \ldots, z^{i,n}$ drawn from $\mathcal{D}_j$, where each data point $z^{i,j}$ consists of a pair of feature and response denoted by $z^{i,\ell} = (x^{i,\ell}, y^{i,\ell})$.

Let $f(\theta; z) : \Theta \to \mathbb{R}$ be the loss function associated with data point z, where $\Theta \subseteq \mathbb{R}^d$ is the parameter space. In this paper, we choose $\Theta = \mathbb{R}^d$. Our goal is to minimize the population loss function $F^j(\theta) := \mathbb{E}_{z \sim \mathcal{D}_j}[f(\theta; z)]$ for all $j \in [k]$. For the purpose of theoretical analysis in Section 5, we focus on the strongly convex losses, in which case we can prove guarantees for estimating the unique solution that minimizes each population loss function. In particular, we try to find solutions $\{\widehat{\theta}_j\}_{j=1}^k$ that are close to $\theta_j^* = \operatorname{argmin}_{\theta \in \Theta} F^j(\theta)$, $j \in [k]$. In our problem, since we only have access to finite data, we take advantage of the empirical loss functions. In particular, let $Z \subseteq \{z^{i,1}, \ldots, z^{i,n}\}$ be a subset of the data points on the i-th machine. We define the empirical loss associated with Z as $F_i(\theta; Z) = \frac{1}{|Z|} \sum_{z \in Z} f(\theta; z)$. When it is clear from the context, we may also use the shorthand notation $F_i(\theta)$ to denote an empirical loss associated with some (or all) data on the i-th worker.

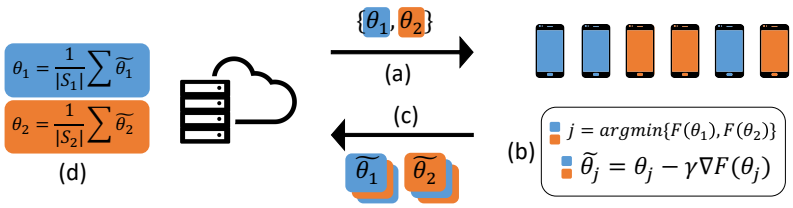


Figure 1: An overview of IFCA (model averaging). (a) The server broadcast models. (b) Worker machines identify their cluster memberships and run local updates. (c) The worker machines send back the local models to server. (d) Average the models within the same estimated cluster S_j.

4 Algorithm

In this section, we provide details of our algorithm. We name this scheme *Iterative Federated Clustering Algorithm* (IFCA). The main idea is to alternatively minimize the loss functions while estimating the cluster identities. We discuss two variations of IFCA, namely gradient averaging and model averaging. The algorithm is formally presented in Algorithm 1 and illustrated in Figure 1.

Algorithm 1: Iterative Federated Clustering Algorithm (IFCA)

Input: number of clusters k, step size γ, $j \in [k]$, initialization $\theta_j^{(0)}$, $j \in [k]$
number of parallel iterations T, number of local gradient steps τ (for model averaging).
for $t = 0, 1, \ldots, T-1$ **do**
  center machine: broadcast $\theta_j^{(t)}$, $j \in [k]$
  $M_t \leftarrow$ random subset of worker machines (participating devices)
  for worker machine $i \in M_t$ in parallel **do**
    cluster identity estimate $\widehat{j} = \text{argmin}_{j \in [k]} F_i(\theta_j^{(t)})$
    define one-hot encoding vector $s_i = \{s_{i,j}\}_{j=1}^k$ with $s_{i,j} = \mathbf{1}\{j = \widehat{j}\}$
    option I (gradient averaging):
      compute (stochastic) gradient: $g_i = \widehat{\nabla} F_i(\theta_{\widehat{j}}^{(t)})$, send back s_i, g_i to the center machine
    option II (model averaging):
      $\widetilde{\theta}_i = \mathsf{LocalUpdate}(\theta_{\widehat{j}}^{(t)}, \gamma, \tau)$, send back s_i, $\widetilde{\theta}_i$ to the center machine
  end for
  center machine:
  option I (gradient averaging): $\theta_j^{(t+1)} = \theta_j^{(t)} - \frac{\gamma}{m} \sum_{i \in M_t} s_{i,j} g_i, \forall\, j \in [k]$
  option II (model averaging): $\theta_j^{(t+1)} = \sum_{i \in M_t} s_{i,j} \widetilde{\theta}_i / \sum_{i \in M_t} s_{i,j}, \forall\, j \in [k]$
end for
return $\theta_j^{(T)}$, $j \in [k]$

$\mathsf{LocalUpdate}(\widetilde{\theta}^{(0)}, \gamma, \tau)$ at the i-th worker machine
for $q = 0, \ldots, \tau - 1$ **do**
  (stochastic) gradient descent $\widetilde{\theta}^{(q+1)} = \widetilde{\theta}^{(q)} - \gamma \widehat{\nabla} F_i(\widetilde{\theta}^{(q)})$
end for
return $\widetilde{\theta}^{(\tau)}$

The algorithm starts with k initial model parameters $\theta_j^{(0)}$, $j \in [k]$. In the t-th iteration of IFCA, the center machine selects a random subset of worker machines, $M_t \subseteq [m]$, and broadcasts the current model parameters $\{\theta_j^{(t)}\}_{j=1}^k$ to the worker machines in M_t. Here, we call M_t the set of *participating devices*. Recall that each worker machine is equipped with local empirical loss function $F_i(\cdot)$. Using the received parameter estimates and F_i, the i-th worker machine ($i \in M_t$) estimates its cluster identity via finding the model parameter with lowest loss, i.e., $\widehat{j} = \text{argmin}_{j \in [k]} F_i(\theta_j^{(t)})$ (ties can be broken arbitrarily). If we choose the option of gradient averaging, the worker machine then computes

a (stochastic) gradient of the local empirical loss F_i at $\theta_{\widehat{j}}^{(t)}$, and sends its cluster identity estimate and gradient back to the center machine. After receiving the gradients and cluster identity estimates from all the participating worker machines, the center machine then collects all the gradient updates from worker machines whose cluster identity estimates are the same and conducts gradient descent update on the model parameter of the corresponding cluster. If we choose the option of model averaging (similar to the Federated Averaging algorithm [26]), each participating device needs to run τ steps of local (stochastic) gradient descent updates, get the updated model, and send the new model and its cluster identity estimate to the center machine. The center machine then averages the new models from the worker machines whose cluster identity estimates are the same.

5 Theoretical guarantees

In this section, we present convergence guarantees of IFCA. In order to streamline our theoretical analysis, we make several simplifications: we consider the IFCA with gradient averaging, and assume that all the worker machines participate in every rounds of IFCA, i.e., $M_t = [m]$ for all t. In addition, we also use the *re-sampling* technique for the purpose of theoretical analysis. In particular, suppose that we run a total of T parallel iterations. We partition the n data points on each machine into $2T$ disjoint subsets, each with $n' = \frac{n}{2T}$ data points. For the i-th machine, we denote the subsets as $\widehat{Z}_i^{(0)}, \ldots, \widehat{Z}_i^{(T-1)}$ and $Z_i^{(0)}, \ldots, Z_i^{(T-1)}$. In the t-th iteration, we use $\widehat{Z}_i^{(t)}$ to estimate the cluster identity, and use $Z_i^{(t)}$ to conduct gradient descent. As we can see, we use fresh data samples for each iteration of the algorithm. Furthermore, in each iteration, we use different set of data points for obtaining the cluster estimate and computing the gradient. This is done in order to remove the inter-dependence between the cluster estimation and the gradient computation, and ensure that in each iteration, we use fresh i.i.d. data that are independent of the current model parameter. We would like to emphasize that re-sampling is a standard tool used in statistics [30, 13, 45, 46, 10], and that it is for theoretical tractability only and is not required in practice as we show in Section 6.

Under these conditions, the update rule for the parameter vector of the j-th cluster can be written as

$$S_j^{(t)} = \{i \in [m] : j = \operatorname{argmin}_{j' \in [k]} F_i(\theta_{j'}^{(t)}; \widehat{Z}_i^{(t)})\},$$
$$\theta_j^{(t+1)} = \theta_j^{(t)} - \frac{\gamma}{m} \sum_{i \in S_j^{(t)}} \nabla F_i(\theta_j^{(t)}; Z_i^{(t)}),$$

where $S_j^{(t)}$ denotes the set of worker machines whose cluster identity estimate is j in the t-th iteration. In the following, we discuss the convergence guarantee of IFCA under two models: in Section 5.1, we analyze the algorithm under a linear model with Gaussian features and squared loss, and in Section 5.2, we analyze the algorithm under a more general setting of strongly convex loss functions.

5.1 Linear models with squared loss

In this section, we analyze our algorithm in a concrete linear model. This model can be seen as a warm-up example for more general problems with strongly convex loss functions that we discuss in Section 5.2, as well as a distributed formulation of the widely studied mixture of linear regression problem [45, 46]. We assume that the data on the worker machines in the j-th cluster are generated in the following way: for $i \in S_j^*$, the feature-response pair of the i-th worker machine machine satisfies

$$y^{i,\ell} = \langle x^{i,\ell}, \theta_j^* \rangle + \epsilon^{i,\ell},$$

where $x^{i,\ell} \sim \mathcal{N}(0, I_d)$ and the additive noise $\epsilon^{i,\ell} \sim \mathcal{N}(0, \sigma^2)$ is independent of $x^{i,\ell}$. Furthermore, we use the squared loss function $f(\theta; x, y) = (y - \langle x, \theta \rangle)^2$. As we can see, this model is the mixture of linear regression model in the distributed setting. We observe that under the above setting, the parameters $\{\theta_j^*\}_{j=1}^k$ are the minimizers of the population loss function $F^j(\cdot)$.

We proceed to analyze our algorithm. We define $p_j := |S_j^*|/m$ as the fraction of worker machines belonging to the j-th cluster, and let $p := \min\{p_1, p_2, \ldots, p_k\}$. We also define the minimum separation Δ as $\Delta := \min_{j \neq j'} \|\theta_j^* - \theta_{j'}^*\|$, and $\rho := \frac{\Delta^2}{\sigma^2}$ as the signal-to-noise ratio. Before we establish our convergence result, we state a few assumptions. Here, recall that n' denotes the number of data that each worker uses in each step.

Assumption 1. *The initialization of parameters $\theta_j^{(0)}$ satisfy $\|\theta_j^{(0)} - \theta_j^*\| \leq \frac{1}{4}\Delta$, $\forall\, j \in [k]$.*

Assumption 2. *Without loss of generality, we assume that $\max_{j\in[k]} \|\theta_j^*\| \lesssim 1$, and that $\sigma \lesssim 1$. We also assume that $n' \gtrsim (\frac{\rho+1}{\rho})^2 \log m$, $d \gtrsim \log m$, $p \gtrsim \frac{\log m}{m}$, $pmn' \gtrsim d$, and $\Delta \gtrsim \frac{\sigma}{p}\sqrt{\frac{d}{mn'}} + \exp(-c(\frac{\rho}{\rho+1})^2 n')$ for some universal constant c.*

In Assumption 1, we assume that the initialization is close enough to θ_j^*. We note that this is a standard assumption in the convergence analysis of mixture models [1, 44], due to the non-convex optimization landscape of mixture model problems. In Assumption 2, we put mild assumptions on n', m, p, and d. The condition that $pmn' \gtrsim d$ simply assumes that the total number of data that we use in each iteration for each cluster is at least as large as the dimension of the parameter space. The condition that $\Delta \gtrsim \frac{\sigma}{p}\sqrt{\frac{d}{mn'}} + \exp(-c(\frac{\rho}{\rho+1})^2 n')$ ensures that the iterates stay close to θ_j^*.

We first provide a single step analysis of our algorithm. We assume that at a certain iteration, we obtain parameter vectors θ_j that are close to the ground truth parameters θ_j^*, and show that θ_j converges to θ_j^* at an exponential rate with an error floor.

Theorem 1. *Consider the linear model and assume that Assumptions 1 and 2 hold. Suppose that in a certain iteration of the IFCA algorithm we obtain parameter vectors θ_j with $\|\theta_j - \theta_j^*\| \leq \frac{1}{4}\Delta$. Let θ_j^+ be iterate after this iteration. Then there exist universal constants $c_1, c_2, c_3, c_4 > 0$ such that when we choose step size $\gamma = c_1/p$, with probability at least $1 - 1/\mathrm{poly}(m)$, we have for all $j \in [k]$,*

$$\|\theta_j^+ - \theta_j^*\| \leq \frac{1}{2}\|\theta_j - \theta_j^*\| + c_2\frac{\sigma}{p}\sqrt{\frac{d}{mn'}} + c_3\exp\left(-c_4(\frac{\rho}{\rho+1})^2 n'\right).$$

We prove Theorem 1 in Appendix A. Here, we briefly summarize the proof idea. Using the initialization condition, we show that the set $\{S_j\}_{j=1}^k$ has a significant overlap with $\{S_j^*\}_{j=1}^k$. In the overlapped set, we then argue that the gradient step provides a contraction and error floor due to the basic properties of linear regression. We then bound the gradient norm of the miss-classified machines and add them to the error floor. We complete the proof by combining the contributions of properly classified and miss-classified worker machines. We can then iteratively apply Theorem 1 and obtain accuracy of the final solution $\widehat{\theta}_j$ in the following corollary.

Corollary 1. *Consider the linear model and assume that Assumptions 1 and 2 hold. By choosing step size $\gamma = c_1/p$, with probability at least $1 - \frac{\log(\Delta/4\varepsilon)}{\mathrm{poly}(m)}$, after $T = \log\frac{\Delta}{4\varepsilon}$ parallel iterations, we have for all $j \in [k]$, $\|\widehat{\theta}_j - \theta_j^*\| \leq \varepsilon$, where $\varepsilon = c_5\frac{\sigma}{p}\sqrt{\frac{d}{mn'}} + c_6\exp(-c_4(\frac{\rho}{\rho+1})^2 n')$.*

Let us examine the final accuracy. Since the number of data points on each worker machine $n = 2n'T = 2n'\log(\Delta/4\varepsilon)$, we know that for the smallest cluster, there are a total of $2pmn'\log(\Delta/4\varepsilon)$ data points. According to the minimax estimation rate of linear regression [40], we know that *even if we know the ground truth cluster identities*, we cannot obtain an error rate better than $\mathcal{O}(\sigma\sqrt{\frac{d}{pmn'\log(\Delta/4\varepsilon)}})$. Comparing this rate with our statistical accuracy ε, we can see that the first term $\frac{\sigma}{p}\sqrt{\frac{d}{mn'}}$ in ε is equivalent to the minimax rate up to a logarithmic factor and a dependency on p, and the second term in ε decays exponentially fast in n', and therefore, our final statistical error rate is *near optimal*.

5.2 Strongly convex loss functions

In this section, we study a more general scenario where the population loss functions of the k clusters are strongly convex and smooth. In contrast to the previous section, our analysis do not rely on any specific statistical model, and thus can be applied to more general machine learning problems. We start with reviewing the standard definitions of strongly convex and smooth functions $F : \mathbb{R}^d \mapsto \mathbb{R}$.

Definition 1. *F is λ-strongly convex if $\forall\theta, \theta'$, $F(\theta') \geq F(\theta) + \langle\nabla F(\theta), \theta' - \theta\rangle + \frac{\lambda}{2}\|\theta' - \theta\|^2$.*

Definition 2. *F is L-smooth if $\forall\theta, \theta'$, $\|\nabla F(\theta) - \nabla F(\theta')\| \leq L\|\theta - \theta'\|$.*

In this section, we assume that the population loss functions $F^j(\theta)$ are strongly convex and smooth.

Assumption 3. *The population loss function $F^j(\theta)$ is λ-strongly convex and L-smooth, $\forall j \in [k]$.*

We note that we do not make any convexity or smoothness assumptions on the individual loss function $f(\theta; z)$. Instead, we make the following distributional assumptions on $f(\theta; z)$ and $\nabla f(\theta; z)$.

Assumption 4. *For every θ and every $j \in [k]$, the variance of $f(\theta; z)$ is upper bounded by η^2, when z is sampled according to $\mathcal{D}_j$, i.e., $\mathbb{E}_{z\sim\mathcal{D}_j}[(f(\theta; z) - F^j(\theta))^2] \leq \eta^2$*

Assumption 5. *For every θ and every $j \in [k]$, the variance of $\nabla f(\theta; z)$ is upper bounded by v^2, when z is sampled according to $\mathcal{D}_j$, i.e., $\mathbb{E}_{z\sim\mathcal{D}_j}[\|\nabla f(\theta; z) - \nabla F^j(\theta)\|_2^2] \leq v^2$*

Bounded variance of gradient is very common in analyzing SGD [5]. In this paper we use loss function value to determine cluster identity, so we also need to have a probabilistic assumption on $f(\theta; z)$. We note that bounded variance is a relatively weak assumption on the tail behavior of probability distributions. In addition to the assumptions above, we still use some definitions from Section 5.1, i.e., $\Delta := \min_{j\neq j'} \|\theta_j^* - \theta_{j'}^*\|$, and $p = \min_{j\in[k]} p_j$ with $p_j = |S_j^*|/m$. We make the following assumptions on the initialization, n', p, and Δ.

Assumption 6. *Without loss of generality, we assume that $\max_{j\in[k]} \|\theta_j^*\| \lesssim 1$. We also assume that $\|\theta_j^{(0)} - \theta_j^*\| \leq \frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta$, $\forall j \in [k]$, $n' \gtrsim \frac{k\eta^2}{\lambda^2\Delta^4}$, $p \gtrsim \frac{\log(mn')}{m}$, and that $\Delta \geq \widetilde{\mathcal{O}}(\max\{(n')^{-1/5}, m^{-1/6}(n')^{-1/3}\})$.*

Here, for simplicity, the $\widetilde{\mathcal{O}}$ notation omits any logarithmic factors and quantities that do not depend on m and n'. As we can see, again we need to assume good initialization, due to the nature of the mixture model, and the assumptions that we impose on n', p, and Δ are relatively mild; in particular, the assumption on Δ ensures that the iterates stay close to an ℓ_2 ball around θ_j^*.

Theorem 2. *Suppose Assumptions 3-6 hold. Choose step size $\gamma = 1/L$. Then, with probability at least $1 - \delta$, after $T = \frac{8L}{p\lambda}\log\left(\frac{\Delta}{2\varepsilon}\right)$ parallel iterations, we have for all $j \in [k]$, $\|\widehat{\theta}_j - \theta_j^*\| \leq \varepsilon$, where*

$$\varepsilon \lesssim \frac{vkL\log(mn')}{p^{5/2}\lambda^2\delta\sqrt{mn'}} + \frac{\eta^2L^2k\log(mn')}{p^2\lambda^4\delta\Delta^4n'} + \widetilde{\mathcal{O}}(\frac{1}{n'\sqrt{m}}).$$

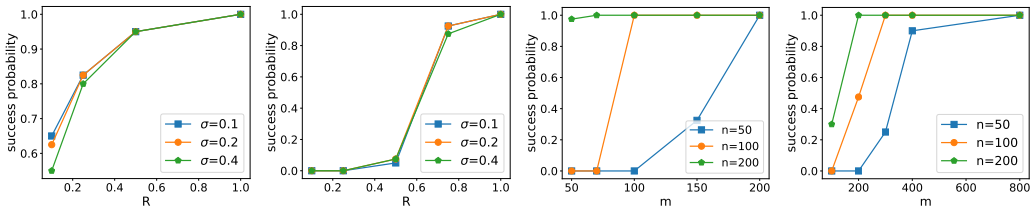
We prove Theorem 2 in the Appendix B. Similar to Section 5.1, to prove this result, we first prove a per-iteration contraction

$$\|\theta_j^+ - \theta_j^*\| \leq (1 - \frac{p\lambda}{8L})\|\theta_j - \theta_j^*\| + \widetilde{\mathcal{O}}(\frac{1}{\sqrt{mn'}} + \frac{1}{n'} + \frac{1}{n'\sqrt{m}}),\ \forall j \in [k],$$

and then derive the convergence rate. To better interpret the result, we focus on the dependency on m and n and treat other quantities as constants. Then, since $n = 2n'T$, we know that n and n' are of the same scale up to a logarithmic factor. Therefore, the final statistical error rate that we obtain is $\epsilon = \widetilde{\mathcal{O}}(\frac{1}{\sqrt{mn}} + \frac{1}{n})$. As discussed in Section 5.1, $\frac{1}{\sqrt{mn}}$ is the optimal rate even if we know the cluster identities; thus our statistical rate is near optimal in the regime where $n \gtrsim m$. In comparison with the statistical rate in linear models $\widetilde{\mathcal{O}}(\frac{1}{\sqrt{mn}} + \exp(-n))$, we note that the major difference is in the second term. The additional terms of the linear model and the strongly convex case are $\exp(-n)$ and $\frac{1}{n}$, respectively. We note that this is due to different statistical assumptions: in for the linear model, we assume Gaussian noise whereas here we only assume bounded variance.

6 Experiments

In this section, we present our experimental results, which not only validate the theoretical claims in Section 5, but also demonstrate that our algorithm can be efficiently applied beyond the regime we discussed in the theory. We emphasize that we *do not* re-sample fresh data points at each iteration. Furthermore, the requirement on the initialization can be relaxed. More specifically, for linear models, we observe that random initialization with a few restarts is sufficient to ensure convergence of Algorithm 1. In our experiments, we also show that our algorithm works efficiently for problems with non-convex loss functions such as neural networks.



(a) $k = 2$ (b) $k = 4$ (c) $k = 2$ (d) $k = 4$

Figure 2: Success probability with respect to: (a), (b) the separation scale R and the scale of additive noise σ; (c), (d) the number of worker machines m and the sample size on each machine n. In (a) and (b), we see that the success probability gets better with increasing R, i.e., more separation between ground truth parameter vectors, and in (c) and (d), we note that the success probability improves with an increase of mn, i.e., more data on each machine and/or more machines.

6.1 Synthetic data

We begin with evaluation of Algorithm 1 with gradient averaging (option I) on linear models with squared loss, as described in Section 5.1. For all $j \in [k]$, we first generate $\theta_j^* \sim \mathsf{Bernoulli}(0.5)$ coordinate-wise, and then rescale their ℓ_2 norm to R. This ensures that the separation between the θ_j^*'s is proportional to R in expectation, and thus, in this experiment, we use R to represent the *separation* between the ground truth parameter vectors. Moreover, we simulate the scenario where all the worker machines participate in all iterations, and all the clusters contain same number of worker machines. For each trial of the experiment, we first generate the parameter vectors θ_j^*'s, fix them, and then randomly initialize $\theta_j^{(0)}$ according to an independent coordinate-wise Bernoulli distribution. We then run Algorithm 1 for 300 iterations, with a constant step size. For $k = 2$ and $k = 4$, we choose the step size in $\{0.01, 0.1, 1\}$, $\{0.5, 1.0, 2.0\}$, respectively. In order to determine whether we successfully learn the model or not, we sweep over the aforementioned step sizes and define the following distance metric: $\mathsf{dist} = \frac{1}{k}\sum_{j=1}^{k} \|\widehat{\theta}_j - \theta_j^*\|$, where $\{\widehat{\theta}_j\}_{j=1}^k$ are the parameter estimates obtained from Algorithm 1. A trial is dubbed *successful* if for a fixed set of θ_j^*, among 10 random initialization of $\theta_j^{(0)}$, at least in one scenario, we obtain $\mathsf{dist} \leq 0.6\sigma$.

In Fig. 2 (a-b), we plot the empirical success probability over 40 trials, with respect to the separation parameter R. We set the problem parameters as (a) $(m, n, d) = (100, 100, 1000)$ with $k = 2$, and (b) $(m, n, d) = (400, 100, 1000)$ with $k = 4$. As we can see, when R becomes larger, i.e., the separation between parameters increases, and the problem becomes easier to solve, yielding in a higher success probability. This validates our theoretical result that higher signal-to-noise ratio produces smaller error floor. In Fig. 2 (c-d), we characterize the dependence on m and n, with fixing R and d with $(R, d) = (0.1, 1000)$ for (c) and $(R, d) = (0.5, 1000)$ for (d). We observe that when we increase m and/or n, the success probability improves. This validates our theoretical finding that more data and/or more worker machines help improve the performance of the algorithm.

6.2 Rotated MNIST and CIFAR

We also create clustered FL datasets based on the MNIST [18] and CIFAR-10 [17] datasets. In order to simulate an environment where the data on different worker machines are generated from different distributions, we augment the datasets using rotation, and create the rotated MNIST [24] and rotated CIFAR datasets. For **rotated MNIST**, recall that the MNIST dataset has 60000 training images and 10000 test images with 10 classes. We first augment the dataset by applying $0, 90, 180, 270$ degrees of rotation to the images, resulting in $k = 4$ clusters. For given m and n satisfying $mn = 60000k$, we randomly partition the images into m worker machines so that each machine holds n images *with the same rotation*. We also split the test data into $m_{\text{test}} = 10000k/n$ worker machines in the same way. The **rotated CIFAR** dataset is also created in a similar way as rotated MNIST, with the main difference being that we create $k = 2$ clusters with 0 and 180 degrees of rotation. We note that creating different tasks by manipulating standard datasets such as MNIST and CIFAR-10 has been widely adopted in the continual learning research community [11, 15, 24]. For clustered FL, creating datasets using rotation helps us simulate a federated learning setup with clear cluster structure.

Table 1: Test accuracies(%) $\pm$ std on rotated MNIST ($k = 4$) and rotated CIFAR ($k = 2$)

	rotated MNIST			rotated CIFAR
m, n	4800, 50	2400, 100	1200, 200	200, 500
IFCA (ours)	**94.20 $\pm$ 0.03**	**95.05 $\pm$ 0.02**	**95.25 $\pm$ 0.40**	**81.51 $\pm$ 1.37**
Global model	86.74 $\pm$ 0.04	88.65 $\pm$ 0.08	89.73 $\pm$ 0.13	77.87 $\pm$ 0.39
Local model	63.32 $\pm$ 0.02	73.66 $\pm$ 0.04	80.05 $\pm$ 0.02	33.97 $\pm$ 1.19

For our MNIST experiments, we use the fully connected neural network with ReLU activations, with a single hidden layer of size 200; and for our CIFAR experiments, we use a convolution neural network model which consists of 2 convolutional layers followed by 2 fully connected layers, and the images are preprocessed by standard data augmentation such as flipping and random cropping.

We compare our IFCA algorithm with two baseline algorithms, i.e., the *global model*, and *local model* schemes. For **IFCA**, we use model averaging (option II in Algorithm 1). For MNIST experiments, we use full worker machines participation ($M_t = [m]$ for all t). For LocalUpdate step in Algorithm 1, we choose $\tau = 10$ and step size $\gamma = 0.1$. For CIFAR experiments, we choose $|M_t| = 0.1m$, and apply step size decay 0.99, and we also set $\tau = 5$ and batch size 50 for LocalUpdate process, following prior works [27]. In the **global model** scheme, the algorithm tries to learn single global model that can make predictions from all the distributions. The algorithm does not consider cluster identities, so model averaging step in Algorithm 1 becomes $\theta^{(t+1)} = \sum_{i \in M_t} \widetilde{\theta}_i / |M_t|$, i.e. averaged over parameters from all the participating machines. In the **local model** scheme, the model in each node performs gradient descent only on local data available, and model averaging is not performed.

For IFCA and the global model scheme, we perform inference in the following way. For every test worker machine, we run inference on all learned models (k models for IFCA and one model for global model scheme), and calculate the accuracy from the model that produces the smallest loss value. For testing the local model baselines, the models are tested by measuring the accuracy on the test data with the same distribution (i.e. those have the same rotation). We report the accuracy averaged over all the models in worker machines. For all algorithms, we run experiment with 5 different random seeds and report the average and standard deviation.

Our experimental results are shown in Table 1. We can observe that our algorithm performs better than the two baselines. As we run the IFCA algorithm, we observe that we can gradually find the underlying cluster identities of the worker machines, and after the correct cluster is found, each model is trained and tested using data with the same distribution, resulting in better accuracy. The global model baseline performs worse than ours since it tries to fit all the data from different distributions, and cannot provide personalized predictions. The local model baseline algorithm overfits to the local data easily, leading to worse performance than ours. We also note that in the concurrent work [25], it has been reported that an algorithm similar to IFCA also outperforms simple baseline such as the global model scheme on FEMNIST dataset [2], but meanwhile several other algorithms outperforms the cluster-based method. We argue that FEMNIST may not have a very clear cluster structure and thus a cluster-based method may not be the best fit. For real-world applications, we suggest using the algorithm that is most suitable for the data distribution.

7 Conclusions and future work

In this paper, we address the clustered FL problem. We propose an iterative algorithm and obtain convergence guarantees for strongly convex and smooth functions. In experiments, we achieve this via random initialization with multiple restarts, and we show that our algorithm works efficiently beyond the convex regime. An immediate future work is to extend the analysis to weakly convex and non-convex functions. Also, the convergence guarantees are local, i.e., a good initialization is required. Obtaining a provable initialization for the clustered FL is an interesting future direction.

Acknowledgement

The authors would like to thank Mehrdad Farajtabar for helpful comments.

References

[1] S. Balakrishnan, M. J. Wainwright, B. Yu, et al. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

[2] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

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

[4] C. Daskalakis, C. Tzamos, and M. Zampetakis. Ten steps of EM suffice for mixtures of two gaussians. *arXiv preprint arXiv:1609.00368*, 2016.

[5] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

[6] W. S. DeSarbo and W. L. Cron. A maximum likelihood methodology for clusterwise linear regression. *Journal of classification*, 5(2):249–282, 1988.

[7] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[8] J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

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

[10] A. Ghosh and K. Ramchandran. Alternating minimization converges super-linearly for mixed linear regression. *arXiv preprint arXiv:2004.10914*, 2020.

[11] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[12] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[13] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674, 2013.

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

[15] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[16] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[17] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] D.-S. Lee. Effective Gaussian mixture learning for video background subtraction. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):827–832, 2005.

[20] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling. RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. *CoRR*, abs/1811.03761, 2018.

[21] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su. Scaling distributed machine learning with the parameter server. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 583–598, 2014.

[22] Q. Li and B. M. Kim. Clustering approach for hybrid recommender system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 33–38. IEEE, 2003.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[24] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

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

[26] B. McMahan and D. Ramage. Federated learning: Collaborative machine learning without centralized training data. `https://research.googleblog.com/2017/04/federated-learning-collaborative.html`, 2017.

[27] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[28] R. P. Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

[29] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. *CoRR*, abs/1902.00146, 2019.

[30] P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems*, pages 2796–2804, 2013.

[31] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

[32] M. Rudelson, R. Vershynin, et al. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.

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

[34] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324, 2002.

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

[36] F. Sattler, S. Wiedemann, K. Müller, and W. Samek. Robust and communication-efficient federated learning from non-iid data. *CoRR*, abs/1903.02891, 2019.

[37] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[38] Y. Sun, S. Ioannidis, and A. Montanari. Learning mixtures of linear classifiers. In *ICML*, pages 721–729, 2014.

[39] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[40] M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[41] I. Waldspurger. Phase retrieval with random gaussian sensing vectors by alternating projections. *IEEE Transactions on Information Theory*, 64(5):3301–3312, 2018.

[42] M. Wedel and W. A. Kamakura. Mixture regression models. In *Market segmentation*, pages 101–124. Springer, 2000.

[43] L. Xu and M. I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[44] B. Yan, M. Yin, and P. Sarkar. Convergence of gradient EM on multi-component mixture of gaussians. In *Advances in Neural Information Processing Systems*, pages 6956–6966, 2017.

[45] X. Yi, C. Caramanis, and S. Sanghavi. Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*, pages 613–621, 2014.

[46] X. Yi, C. Caramanis, and S. Sanghavi. Solving a mixture of many random linear equations by tensor decomposition and alternating minimization. *arXiv preprint arXiv:1608.05749*, 2016.

[47] D. Yin, R. Pedarsani, Y. Chen, and K. Ramchandran. Learning mixtures of sparse linear regressions using sparse graph codes. *IEEE Transactions on Information Theory*, 65(3):1430–1451, 2018.

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

[49] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pages 2595–2603, 2010.

Appendix

In our proofs, we use $c, c_1, c_2, \ldots$ to denote positive universal constants, the value of which may differ across instances. For a matrix A, we write $\|A\|_{op}$ and $\|A\|_F$ as the operator norm and Frobenius norm, respectively. For a set S, we use $\overline{S}$ to denote the complement of the set.

A Proof of Theorem 1

Since we only analyze a single iteration, for simplicity we drop the superscript that indicates the iteration counter. Suppose that at a particular iteration, we have model parameters θ_j, $j \in [k]$, for the k clusters. We denote the *estimation* of the set of worker machines that belongs to the j-th cluster by S_j, and recall that the true clusters are denoted by S_j^*, $j \in [k]$.

Probability of erroneous cluster identity estimation We begin with the analysis of the probability of incorrect cluster identity estimation. Suppose that a worker machine i belongs to S_j^*. We define the event $\mathcal{E}_i^{j,j'}$ as the event when the i-th machine is classified to the j'-th cluster, i.e., $i \in S_{j'}$. Thus the event that worker i is correctly classified is $\mathcal{E}_i^{j,j}$, and we use the shorthand notation $\mathcal{E}_i := \mathcal{E}_i^{j,j}$. We now provide the following lemma that bounds the probability of $\mathcal{E}_i^{j,j'}$ for $j' \neq j$.

Lemma 1. *Suppose that worker machine $i \in S_j^*$. Let $\rho := \frac{\Delta^2}{\sigma^2}$. Then there exist universal constants c_1 and c_2 such that for any $j' \neq j$,*

$$\mathbb{P}(\mathcal{E}_i^{j,j'}) \leq c_1 \exp\left(-c_2 n' (\frac{\rho}{\rho+1})^2\right),$$

and by union bound

$$\mathbb{P}(\overline{\mathcal{E}_i}) \leq c_1 k \exp\left(-c_2 n' (\frac{\rho}{\rho+1})^2\right).$$

We prove Lemma 1 in Appendix A.1.

Now we proceed to analyze the gradient descent step. Without loss of generality, we only analyze the first cluster. The update rule of θ_1 in this iteration can be written as

$$\theta_1^+ = \theta_1 - \frac{\gamma}{m} \sum_{i \in S_1} \nabla F_i(\theta_1; Z_i),$$

where Z_i is the set of the n' data points that we use to compute gradient in this iteration on a particular worker machine.

We use the shorthand notation $F_i(\theta) := F_i(\theta; Z_i)$, and note that $F_i(\theta)$ can be written in the matrix form as

$$F_i(\theta) = \frac{1}{n'} \|Y_i - X_i \theta\|^2,$$

where we have the feature matrix $X_i \in \mathbb{R}^{n' \times d}$ and response vector $Y_i = X_i \theta_1^* + \epsilon_i$. According to our model, all the entries of X_i are i.i.d. sampled according to $\mathcal{N}(0,1)$, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$.

We first notice that

$$\|\theta_1^+ - \theta_1^*\| = \| \underbrace{\theta_1 - \theta_1^* - \frac{\gamma}{m} \sum_{i \in S_1 \cap S_1^*} \nabla F_i(\theta_1)}_{T_1} - \underbrace{\frac{\gamma}{m} \sum_{i \in S_1 \cap \overline{S_1^*}} \nabla F_i(\theta_1)}_{T_2} \| \leq \|T_1\| + \|T_2\|.$$

We control the two terms separately. Let us first focus on $\|T_1\|$.

Bound $\|T_1\|$ To simplify notation, we concatenate all the feature matrices and response vectors of all the worker machines in $S_1 \cap S_1^*$ and get the new feature matrix $X \in \mathbb{R}^{N \times d}$, $Y \in \mathbb{R}^N$ with

$Y = X\theta_1^* + \epsilon$, where $N := n'|S_1 \cap S_1^*|$. It is then easy to verify that

$$\begin{aligned} T_1 &= (I - \frac{2\gamma}{mn'}X^\top X)(\theta_1 - \theta_1^*) + \frac{2\gamma}{mn'}X^\top \epsilon \\ &= (I - \frac{2\gamma}{mn'}\mathbb{E}[X^\top X])(\theta_1 - \theta_1^*) + \frac{2\gamma}{mn'}(\mathbb{E}[X^\top X] - X^\top X)(\theta_1 - \theta_1^*) + \frac{2\gamma}{mn'}X^\top \epsilon \\ &= (1 - \frac{2\gamma N}{mn'})(\theta_1 - \theta_1^*) + \frac{2\gamma}{mn'}(\mathbb{E}[X^\top X] - X^\top X)(\theta_1 - \theta_1^*) + \frac{2\gamma}{mn'}X^\top \epsilon. \end{aligned}$$

Therefore

$$\|T_1\| \leq (1 - \frac{2\gamma N}{mn'})\|\theta_1 - \theta_1^*\| + \frac{2\gamma}{mn'}\|X^\top X - \mathbb{E}[X^\top X]\|_{op}\|\theta_1 - \theta_1^*\| + \frac{2\gamma}{mn'}\|X^\top \epsilon\|. \quad (1)$$

Thus in order to bound $\|T_1\|$, we need to analyze two terms, $\|X^\top X - \mathbb{E}[X^\top X]\|_{op}$ and $\|X^\top \epsilon\|$. To bound $\|X^\top X - \mathbb{E}[X^\top X]\|_{op}$, we first provide an analysis of N showing that it is large enough. Using Lemma 1 in conjunction with Assumption 2, we see that the probability of correctly classifying any worker machine i, given by $\mathbb{P}(\mathcal{E}_i)$, satisfies $\mathbb{P}(\mathcal{E}_i) \geq \frac{1}{2}$. Hence, we obtain

$$\mathbb{E}[|S_1 \cap S_1^*|] \geq \mathbb{E}[\frac{1}{2}|S_1^*|] = \frac{1}{2}p_1 m,$$

where we use the fact that $|S_1^*| = p_1 m$. Since $|S_1 \cap S_1^*|$ is a sum of Bernoulli random variables with success probability at least $\frac{1}{2}$, we obtain

$$\mathbb{P}\left(|S_1 \cap S_1^*| \leq \frac{1}{4}p_1 m\right) \leq \mathbb{P}\left(\left||S_1 \cap S_1^*| - \mathbb{E}[|S_1 \cap S_1^*|]\right| \geq \frac{1}{4}p_1 m\right) \leq 2\exp(-cpm),$$

where $p = \min\{p_1, p_2, \ldots, p_k\}$, and the second step follows from Hoeffding's inequality. Hence, we obtain $|S_1 \cap S_1^*| \geq \frac{1}{4}p_1 m$ with high probability, which yields

$$\mathbb{P}(N \geq \frac{1}{4}p_1 mn') \geq 1 - 2\exp(-cpm). \quad (2)$$

By combining this fact with our assumption that $pmn' \gtrsim d$, we know that $N \gtrsim d$. Then, according to the concentration of the covariance of Gaussian random vectors [40], we know that with probability at least $1 - 2\exp(-\frac{1}{2}d)$,

$$\|X^\top X - \mathbb{E}[X^\top X]\|_{op} \leq 6\sqrt{dN} \lesssim N. \quad (3)$$

We now proceed to bound $\|X^\top \epsilon\|$. In particular, we use the following lemma.

Lemma 2. *Consider a random matrix $X \in \mathbb{R}^{N \times d}$ with i.i.d. entries sampled according to $\mathcal{N}(0, 1)$, and $\epsilon \in \mathbb{R}^N$ be a random vector sampled according to $\mathcal{N}(0, \sigma^2 I)$, independently of X. Then we have with probability at least $1 - 2\exp(-c_1 \max\{d, N\})$,*

$$\|X\|_{op} \leq c\max\{\sqrt{d}, \sqrt{N}\},$$

and with probability at least $1 - c_2\exp(-c_3 \min\{d, N\})$,

$$\|X^\top \epsilon\| \leq c_4 \sigma\sqrt{dN}.$$

We prove Lemma 2 in Appendix A.2. Now we can combine (1), (3), (2), and Lemma 2 and obtain with probability at least $1 - c_1\exp(-c_2 pm) - c_3\exp(-c_4 d)$,

$$\|T_1\| \leq (1 - c_5\gamma p)\|\theta_1 - \theta_1^*\| + c_6\gamma\sigma\sqrt{\frac{d}{mn'}}. \quad (4)$$

Since we assume that $p \gtrsim \frac{\log m}{m}$ and $d \gtrsim \log m$, the success probability can be simplified as $1 - 1/\mathrm{poly}(m)$.

Bound $\|T_2\|$ We first condition on S_1. We have the following:

$$\nabla F_i(\theta_1) = \frac{2}{n'} X_i^\top (Y_i - X_i\theta_1).$$

For $i \in S_1 \cap S_j^*$, with $j \neq 1$, we have $Y_i = X_i\theta_j^* + \epsilon_i$, and so we obtain

$$n'\nabla F_i(\theta_1) = 2X_i^\top X_i(\theta_j^* - \theta_1) + 2X_i^\top \epsilon_i,$$

which yields

$$n'\|\nabla F_i(\theta_1)\| \lesssim \|X_i\|_{op}^2 + \|X_i^\top \epsilon_i\|, \tag{5}$$

where we use the fact that $\|\theta_j^* - \theta_1\| \leq \|\theta_j^*\| + \|\theta_1^*\| + \|\theta_1^* - \theta_1\| \lesssim 1$. Then, we combine (5) and Lemma 2 and get with probability at least $1 - c_1\exp(-c_2\min\{d, n'\})$,

$$\|\nabla F_i(\theta_1)\| \leq \frac{1}{n'}(c_3\max\{d, n'\} + c_4\sigma\sqrt{dn'}) \leq c_5\max\{1, \frac{d}{n'}\}, \tag{6}$$

where we use our assumption that $\sigma \lesssim 1$. By union bound, we know that with probability at least $1 - c_1 m\exp(-c_2\min\{d, n'\})$, (6) holds for all $j \in \overline{S_1^*}$. In addition, since we assume that $n' \gtrsim \log m$, $d \gtrsim \log m$, this probability can be lower bounded by $1 - 1/\mathrm{poly}(m)$. This implies that conditioned on S_1, with probability at least $1 - 1/\mathrm{poly}(m)$,

$$\|T_2\| \leq c_5\frac{\gamma}{m}|S_1 \cap \overline{S_1^*}|\max\{1, \frac{d}{n'}\}. \tag{7}$$

Since we choose $\gamma = \frac{c}{p}$, we have $\frac{\gamma}{m}\max\{1, \frac{d}{n'}\} \lesssim 1$, where we use our assumption that $pmn' \gtrsim d$. This shows that with probability at least $1 - 1/\mathrm{poly}(m)$,

$$\|T_2\| \leq c_5|S_1 \cap \overline{S_1^*}|. \tag{8}$$

We then analyze $|S_1 \cap \overline{S_1^*}|$. By Lemma 1, we have

$$\mathbb{E}[|S_1 \cap \overline{S_1^*}|] \leq c_6 m\exp(-c_7(\frac{\rho}{\rho+1})^2 n'). \tag{9}$$

According to Assumption 2, we know that $n' \geq c(\frac{\rho+1}{\rho})^2\log m$, for some constant c that is large enough. Therefore, $m \leq \exp(\frac{1}{c}(\frac{\rho}{\rho+1})^2 n')$, and thus, as long as c is large enough such that $\frac{1}{c} < c_7$ where c_7 is defined in (9), we have

$$\mathbb{E}[|S_1 \cap \overline{S_1^*}|] \leq c_6\exp(-c_8(\frac{\rho}{\rho+1})^2 n'). \tag{10}$$

and then by Markov's inequality, we have

$$\mathbb{P}\left(|S_1 \cap \overline{S_1^*}| \leq c_6\exp(-\frac{c_8}{2}(\frac{\rho}{\rho+1})^2 n')\right) \geq 1 - \exp(-\frac{c_8}{2}(\frac{\rho}{\rho+1})^2 n')) \geq 1 - \mathrm{poly}(m). \tag{11}$$

Combining (8) with (11), we know that with probability at least $1 - 1/\mathrm{poly}(m)$,

$$\|T_2\| \leq c_1\exp(-c_2(\frac{\rho}{\rho+1})^2 n').$$

Using this fact and (4), we obtain that with probability at least $1 - 1/\mathrm{poly}(m)$,

$$\|\theta_1^+ - \theta_1^*\| \leq (1 - c_1\gamma p)\|\theta_1 - \theta_1^*\| + c_2\gamma\sigma\sqrt{\frac{d}{mn'}} + c_3\exp(-c_4(\frac{\rho}{\rho+1})^2 n').$$

Then we can complete the proof for the first cluster by choosing $\gamma = \frac{1}{2c_1 p}$. To complete the proof for all the k clusters, we can use union bound, and the success probability is $1 - k/\mathrm{poly}(m)$. However, since $k \leq m$ by definition, we still have success probability $1 - 1/\mathrm{poly}(m)$.

A.1 Proof of Lemma 1

Without loss of generality, we analyze $\mathcal{E}_i^{1,j}$ for some $j \neq 1$. By definition, we have

$$\mathcal{E}_i^{1,j} = \{F_i(\theta_j; \widehat{Z}_i) \leq F_i(\theta_1; \widehat{Z}_i)\},$$

where $\widehat{Z}_i$ is the set of n' data points that we use to estimate the cluster identity in this iteration. We write the data points in $\widehat{Z}_i$ in matrix form with feature matrix $X_i \in \mathbb{R}^{n' \times d}$ and response vector $Y_i = X_i\theta_1^* + \epsilon_i$. According to our model, all the entries of X_i are i.i.d. sampled according to $\mathcal{N}(0,1)$, and $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$. Then, we have

$$\mathbb{P}\{\mathcal{E}_i^{1,j}\} = \mathbb{P}\left\{\|X_i(\theta_1^* - \theta_1) + \epsilon_i\|^2 \geq \|X_i(\theta_1^* - \theta_j) + \epsilon_i\|^2\right\}.$$

Consider the random vector $X_i(\theta_1^* - \theta_j) + \epsilon_i$, and in particular consider the ℓ-th coordinate of it. Since X_i and ϵ_i are independent and we resample (X_i, Y_i) at each iteration, the ℓ-th coordinate of $X_i(\theta_1^* - \theta_j) + \epsilon_i$ is a Gaussian random variable with mean 0 and variance $\|\theta_j - \theta_1^*\|^2 + \sigma^2$. Since X_i and ϵ_i contain independent rows, the distribution of $\|X_i(\theta_1^* - \theta_j) + \epsilon_i\|^2$ is given by $(\|\theta_j - \theta_1^*\|^2 + \sigma^2)u_j$, where u_j is a standard Chi-squared random variable n' degrees of freedom. We now calculate the an upper bound on the following probability:

$$\begin{aligned}
&\mathbb{P}\left\{\|X_i(\theta_1^* - \theta_1) + \epsilon_i\|^2 \geq \|X_i(\theta_1^* - \theta_j) + \epsilon_i\|^2\right\} \\
\overset{\text{(i)}}{\leq}&\mathbb{P}\left\{\|X_i(\theta_1^* - \theta_j) + \epsilon_i\|^2 \leq t\right\} + \mathbb{P}\left\{\|X_i(\theta_1^* - \theta_1) + \epsilon_i\|^2 > t\right\} \\
\leq&\mathbb{P}\left\{(\|\theta_j - \theta_1^*\|^2 + \sigma^2)u_j \leq t\right\} + \mathbb{P}\left\{(\|\theta_1 - \theta_1^*\|^2 + \sigma^2)u_1 > t\right\},
\end{aligned} \tag{12}$$

where (i) holds for all $t \geq 0$. For the first term, we use the concentration property of Chi-squared random variables. Using the fact that $\|\theta_j - \theta_1^*\| \geq \|\theta_j^* - \theta_1^*\| - \|\theta_j - \theta_j^*\| \geq \frac{3}{4}\Delta$, we have

$$\mathbb{P}\left\{(\|\theta_j - \theta_1^*\|^2 + \sigma^2)u_j \leq t\right\} \leq \mathbb{P}\left\{(\frac{9}{16}\Delta^2 + \sigma^2)u_j \leq t\right\}. \tag{13}$$

Similarly, using the initialization condition, $\|\theta_1 - \theta_1^*\| \leq \frac{1}{4}\Delta$, the second term of equation (12) can be simplified as

$$\mathbb{P}\left\{(\|\theta_1 - \theta_1^*\|^2 + \sigma^2)u_1 > t\right\} \leq \mathbb{P}\left\{(\frac{1}{16}\Delta^2 + \sigma^2)u_1 > t\right\}. \tag{14}$$

Based on the above observation, we now choose $t = n'(\frac{5}{16}\Delta^2 + \sigma^2)$. Recall that $\rho := \frac{\Delta^2}{\sigma^2}$. Then the inequlity (13) can be rewritten as

$$\mathbb{P}\left\{(\|\theta_j - \theta_1^*\|^2 + \sigma^2)u_j \leq t\right\} \leq \mathbb{P}\left\{\frac{u_j}{n'} - 1 \leq -\frac{4\rho}{9\rho + 16}\right\}.$$

According to the concentration results for standard Chi-squared distribution [40], we know that there exists universal constants c_1 and c_2 such that

$$\mathbb{P}\left\{(\|\theta_j - \theta_1^*\|^2 + \sigma^2)u_j \leq t\right\} \leq c_1 \exp\left(-c_2 n'(\frac{\rho}{\rho+1})^2\right). \tag{15}$$

Similarly, the inequality (14) can be rewritten as

$$\mathbb{P}\left\{(\|\theta_1 - \theta_1^*\|^2 + \sigma^2)u_1 > t\right\} \leq \mathbb{P}\left\{\frac{u_1}{n'} - 1 > \frac{4\rho}{\rho + 16},\right\}$$

and again according to the concentration of Chi-squared distribution, there exists universal constants c_3 and c_4 such that

$$\mathbb{P}\left\{(\|\theta_1 - \theta_1^*\|^2 + \sigma^2)u_1 > t\right\} \leq c_3 \exp\left(-c_4 n'(\frac{\rho}{\rho+1})^2\right). \tag{16}$$

The proof can be completed by combining (12), (15) and (16).

A.2 Proof of Lemma 2

According to Theorem 5.39 of [39], we have with probability at least $1 - 2\exp(-c_1 \max\{d, N\})$,

$$\|X\|_{op} \leq c \max\{\sqrt{d}, \sqrt{N}\},$$

where c and c_1 are universal constants. As for $\|X^\top \epsilon\|$, we first condition on X. According to the Hanson-Wright inequality [32], we obtain for every $t \geq 0$

$$\mathbb{P}\left(\left|\|X^\top \epsilon\| - \sigma \|X^\top\|_F\right| > t\right) \leq 2\exp\left(-c\frac{t^2}{\sigma^2 \|X^\top\|_{op}^2}\right). \tag{17}$$

Using Chi-squared concentration [40], we obtain with probability at least $1 - 2\exp(-cdN)$,

$$\|X\|_F \leq c\sqrt{dN}.$$

Furthermore, using the fact that $\|X^\top\|_{op} = \|X\|_{op}$ and substituting $t = \sigma\sqrt{dN}$ in (17), we obtain with probability at least $1 - c_2 \exp(-c_3 \min\{d, N\})$,

$$\|X^\top \epsilon\| \leq c_4 \sigma \sqrt{dN}.$$

B Proof of Theorem 2

The proof of this theorem is similar to that of the linear model. We begin with a single-step analysis.

B.1 Analysis for a single step

Suppose that at a certain step, we have model parameters θ_j, $j \in [k]$ for the k clusters. Assume that $\|\theta_j - \theta_j^*\| \leq \frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta$, for all $j \in [k]$.

Probability of erroneous cluster identity estimation: We first calculate the probability of erroneous estimation of worker machines' cluster identity. We define the events $\mathcal{E}_i^{j,j'}$ in the same way as in Appendix A, and have the following lemma.

Lemma 3. *Suppose that worker machine $i \in S_j^*$. Then there exists a universal constants c_1 such that for any $j' \neq j$,*

$$\mathbb{P}(\mathcal{E}_i^{j,j'}) \leq c_1 \frac{\eta^2}{\lambda^2 \Delta^4 n'},$$

and by union bound

$$\mathbb{P}(\overline{\mathcal{E}_i}) \leq c_1 \frac{k\eta^2}{\lambda^2 \Delta^4 n'}.$$

We prove Lemma 3 in Appendix B.3. Now we proceed to analyze the gradient descent iteration. Without loss of generality, we focus on θ_1. We have

$$\|\theta_1^+ - \theta_1^*\| = \|\theta_1 - \theta_1^* - \frac{\gamma}{m}\sum_{i \in S_1} \nabla F_i(\theta_1)\|,$$

where $F_i(\theta) := F_i(\theta; Z_i)$ with Z_i being the set of data points on the i-th worker machine that we use to compute the gradient, and S_1 is the set of indices returned by Algorithm 1 corresponding to the first cluster. Since

$$S_1 = (S_1 \cap S_1^*) \cup (S_1 \cap \overline{S_1^*})$$

and the sets are disjoint, we have

$$\|\theta_1^+ - \theta_1^*\| = \|\underbrace{\theta_1 - \theta_1^* - \frac{\gamma}{m}\sum_{i \in S_1 \cap S_1^*} \nabla F_i(\theta_1)}_{T_1} - \underbrace{\frac{\gamma}{m}\sum_{i \in S_1 \cap \overline{S_1^*}} \nabla F_i(\theta_1)}_{T_2}\|.$$

Using triangle inequality, we obtain

$$\|\theta_1^+ - \theta_1^*\| \leq \|T_1\| + \|T_2\|,$$

and we control both the terms separately. Let us first focus on $\|T_1\|$.

Bound $\|T_1\|$ We first split T_1 in the following way:

$$T_1 = \underbrace{\theta_1 - \theta_1^* - \widehat{\gamma}\nabla F^1(\theta_1)}_{T_{11}} + \underbrace{\widehat{\gamma}\big(\nabla F^1(\theta_1) - \frac{1}{|S_1 \cap S_1^*|}\sum_{i\in S_1\cap S_1^*}\nabla F_i(\theta_1)\big)}_{T_{12}}, \tag{18}$$

where $\widehat{\gamma} := \gamma\frac{|S_1\cap S_1^*|}{m}$. Let us condition on S_1. According to standard analysis technique for gradient descent on strongly convex functions, we know that when $\widehat{\gamma} \le \frac{1}{L}$,

$$\|T_{11}\| = \|\theta_1 - \theta_1^* - \widehat{\gamma}\nabla F^1(\theta_1)\| \le (1 - \frac{\widehat{\gamma}\lambda L}{\lambda + L})\|\theta_1 - \theta_1^*\|. \tag{19}$$

Further, we have $\mathbb{E}[\|T_{12}\|^2] = \frac{v^2}{n'|S_1\cap S_1^*|}$, which implies $\mathbb{E}[\|T_{12}\|] \le \frac{v}{\sqrt{n'|S_1\cap S_1^*|}}$, and thus by Markov's inequality, for any $\delta_0 > 0$, with probability at least $1 - \delta_0$,

$$\|T_{12}\| \le \frac{v}{\delta_0\sqrt{n'|S_1 \cap S_1^*|}}. \tag{20}$$

We then analyze $|S_1 \cap S_1^*|$. Similar to the proof of Theorem 1, we can show that $|S_1 \cap S_1^*|$ is large enough. From Lemma 3 and using our assumption, we see that the probability of correctly classifying any worker machine i, given by $\mathbb{P}(\mathcal{E}_i)$, satisfies $\mathbb{P}(\mathcal{E}_i) \ge \frac{1}{2}$. Recall $p = \min\{p_1, p_2, \ldots, p_k\}$, and we obtain $|S_1 \cap S_1^*| \ge \frac{1}{4}p_1 m$ with probability at least $1 - 2\exp(-cpm)$. Let us condition on $|S_1 \cap S_1^*| \ge \frac{1}{4}p_1 m$ and choose $\gamma = 1/L$. Then $\widehat{\gamma} \le 1/L$ is satisfied, and on the other hand $\widehat{\gamma} \ge \frac{p}{4L}$. Plug this fact in (19), we obtain

$$\|T_{11}\| \le (1 - \frac{p\lambda}{8L})\|\theta_1 - \theta_1^*\|. \tag{21}$$

We then combine (20) and (21) and have with probability at least $1 - \delta_0 - 2\exp(-cpm)$,

$$\|T_1\| \le (1 - \frac{p\lambda}{8L})\|\theta_1 - \theta_1^*\| + \frac{2v}{\delta_0 L\sqrt{pmn'}}. \tag{22}$$

Bound $\|T_2\|$ Let us define $T_{2j} := \sum_{S_1\cap S_j^*}\nabla F_i(\theta_1)$, $j \ge 2$. We have $T_2 = \frac{\gamma}{m}\sum_{j=2}^k T_{2j}$. We condition on S_1 and first analyze T_{2j}. We have

$$T_{2j} = |S_1 \cap S_j^*|\nabla F^j(\theta_1) + \sum_{i\in S_1\cap S_j^*}\left(\nabla F_i(\theta_1) - \nabla F^j(\theta_1)\right). \tag{23}$$

Due to the smoothness of $F^j(\theta)$, we know that

$$\|\nabla F^j(\theta_1)\| \le L\|\theta_1 - \theta_j^*\| \le 3L, \tag{24}$$

where we use the fact that $\|\theta_1 - \theta_j^*\| \le \|\theta_j^*\| + \|\theta_1^*\| + \|\theta_1 - \theta_1^*\| \le 1 + 1 + \frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta \le 3$. In addition, we have

$$\mathbb{E}\left[\left\|\sum_{i\in S_1\cap S_j^*}\nabla F_i(\theta_1) - \nabla F^j(\theta_1)\right\|^2\right] = |S_1 \cap S_j^*|\frac{v^2}{n'},$$

which implies

$$\mathbb{E}\left[\left\|\sum_{i\in S_1\cap S_j^*}\nabla F_i(\theta_1) - \nabla F^j(\theta_1)\right\|\right] \le \sqrt{|S_1 \cap S_j^*|}\frac{v}{\sqrt{n'}},$$

and then according to Markov's inequality, for any $\delta_1 \in (0, 1)$, with probability at least $1 - \delta_1$,

$$\left\|\sum_{i\in S_1\cap S_j^*}\nabla F_i(\theta_1) - \nabla F^j(\theta_1)\right\| \le \sqrt{|S_1 \cap S_j^*|}\frac{v}{\delta_1\sqrt{n'}}. \tag{25}$$

Then, by combining (24) and (25), we know that with probability at least $1-\delta_1$,

$$\|T_{2j}\| \le 3L|S_1 \cap S_j^*| + \sqrt{|S_1 \cap S_j^*|}\frac{v}{\delta_1\sqrt{n'}}. \tag{26}$$

By union bound, we know that with probability at least $1-k\delta_1$, (26) applies to all $j \neq 1$. Then, we have with probability at least $1-k\delta_1$,

$$\|T_2\| \le \frac{3\gamma L}{m}|S_1 \cap \overline{S_1^*}| + \frac{\gamma v\sqrt{k}}{\delta_1 m\sqrt{n'}}\sqrt{|S_1 \cap \overline{S_1^*}|}. \tag{27}$$

According to Lemma 3, we know that

$$\mathbb{E}[|S_1 \cap \overline{S_1^*}|] \le c_1\frac{\eta^2 m}{\lambda^2\Delta^4 n'}.$$

Then by Markov's inequality, we know that with probability at least $1-\delta_2$,

$$|S_1 \cap \overline{S_1^*}| \le c_1\frac{\eta^2 m}{\delta_2\lambda^2\Delta^4 n'}. \tag{28}$$

Now we combine (27) with (28) and obtain with probability at least $1-k\delta_1-\delta_2$,

$$\|T_2\| \le c_1\frac{\eta^2}{\delta_2\lambda^2\Delta^4 n'} + c_2\frac{v\eta\sqrt{k}}{\delta_1\sqrt{\delta_2}\lambda L\Delta^2\sqrt{mn'}}. \tag{29}$$

Combining (22) and (29), we know that with probability at least $1-\delta_0-k\delta_1-\delta_2-2\exp(-cpm)$,

$$\|\theta_1^+ - \theta_1^*\| \le (1-\frac{p\lambda}{8L})\|\theta_1-\theta_1^*\| + \frac{2v}{\delta_0 L\sqrt{pmn'}} + c_1\frac{\eta^2}{\delta_2\lambda^2\Delta^4 n'} + c_2\frac{v\eta\sqrt{k}}{\delta_1\sqrt{\delta_2}\lambda L\Delta^2\sqrt{mn'}}. \tag{30}$$

In the following, we let $\delta_3 := \delta_0 + k\delta_1 + \delta_2 + 2\exp(-cpm)$, and

$$\varepsilon_0 = \frac{2v}{\delta_0 L\sqrt{pmn'}} + c_1\frac{\eta^2}{\delta_2\lambda^2\Delta^4 n'} + c_2\frac{v\eta\sqrt{k}}{\delta_1\sqrt{\delta_2}\lambda L\Delta^2\sqrt{mn'}}.$$

Let us simplify this expression. We first choose $\delta \in (0,1)$ as the failure probability of the entire algorithm. Then, we choose

$$\delta_0 = \frac{p\lambda\delta}{CkL\log(mn')}, \quad \delta_1 = \frac{p\lambda\delta}{Ck^2L\log(mn')}, \quad \delta_2 = \frac{p\lambda\delta}{CkL\log(mn')},$$

for some constant $C>0$ that is large enough. In addition, since we assume that $p \gtrsim \frac{\log(mn')}{m}$, we have $\exp(-cpm) \le 1/\mathrm{poly}(mn') \lesssim \frac{p\lambda\delta}{kL\log(mn')}$. Consider all these facts, we obtain

$$\delta_3 = \frac{4p\lambda\delta}{CkL\log(mn')}, \tag{31}$$

$$\varepsilon_0 \lesssim \frac{vk\log(mn')}{p^{3/2}\lambda\delta\sqrt{mn'}} + \frac{\eta^2 Lk\log(mn')}{p\lambda^3\delta\Delta^4 n'} + \frac{v\eta k^3\sqrt{L}\log^{3/2}(mn')}{p^{3/2}\lambda^{5/2}\delta^{3/2}\Delta^2\sqrt{mn'}}. \tag{32}$$

In addition, by union bound, we know that with probability at least $1-k\delta_3$, for all $j\in[k]$,

$$\|\theta_j^+ - \theta_j^*\| \le (1-\frac{p\lambda}{8L})\|\theta_j-\theta_j^*\| + \varepsilon_0. \tag{33}$$

B.2 Convergence of the algorithm

We now analyze the convergence of the entire algorithm. First, we can verify that as long as

$$\varepsilon_0 \le \frac{p}{32}(\frac{\lambda}{L})^{3/2}\Delta, \tag{34}$$

we can guarantee that $\|\theta_j^+ - \theta_j^*\| \le \frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta$. We can also verify that as long as there is

$$\Delta \ge \widetilde{\mathcal{O}}(\max\{(n')^{-1/5}, m^{-1/6}(n')^{-1/3}\}), \tag{35}$$

using the definition of ε_0 in (32), we know that (34) holds. Here, in the $\widetilde{\mathcal{O}}$ notation, we omit the logarithmic factors and quantities that does not depend on m and n'. In this case, we can iteratively apply (33) for T iterations and obtain that with probability at least $1 - kT\delta_3$,

$$\|\theta_j^{(T)} - \theta_j^*\| \leq (1 - \frac{p\lambda}{8L})^T \|\theta_j^{(0)} - \theta_j^*\| + \frac{8L}{p\lambda}\varepsilon_0.$$

Then, we know that when we choose

$$T = \frac{8L}{p\lambda} \log\left(\frac{p\lambda\Delta}{32\varepsilon_0 L}\right), \tag{36}$$

we have

$$(1 - \frac{p\lambda}{8L})^T \|\theta_j^{(0)} - \theta_j^*\| \leq \exp(-\frac{p\lambda}{8L}T)\frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta \leq \frac{8}{p}\sqrt{\frac{L}{\lambda}}\varepsilon_0,$$

which implies $\|\theta_j^{(T)} - \theta_j^*\| \leq \frac{16L}{p\lambda}\varepsilon_0$. Finally, we check the failure probability. The failure probability is

$$kT\delta_3 \leq \frac{8kL}{p\lambda}\log\left(\frac{p\lambda\Delta}{32\varepsilon_0 L}\right)\frac{4p\lambda\delta}{CkL\log(mn')} = \frac{32\delta}{C}\frac{\log(\frac{p\lambda\Delta}{32\varepsilon_0 L})}{\log(mn')} \leq \delta\frac{\log(\frac{1}{\varepsilon_0})}{\log((mn')^{C/32})}.$$

On the other hand, according to (32), we know that

$$\frac{1}{\varepsilon_0} \leq \widetilde{\mathcal{O}}(\max\{\sqrt{mn'}, n'\}),$$

then, as long as C is large enough, we can guarantee that $(mn')^{C/32} > \frac{1}{\varepsilon_0}$, which implies that the failure probability is upper bounded by δ. Our final error floor can be obtained by redefining

$$\varepsilon := \frac{16L}{p\lambda}\varepsilon_0.$$

B.3 Proof of Lemma 3

Without loss of generality, we bound the probability of $\mathcal{E}_i^{1,j}$ for some $j \neq 1$. We know that

$$\mathcal{E}_i^{1,j} = \left\{F_i(\theta_1; \widehat{Z}_i) \geq F_i(\theta_j; \widehat{Z}_i)\right\},$$

where $\widehat{Z}_i$ is the set of n' data points that we use to estimate the cluster identity in this iteration. In the following, we use the shorthand notation $F_i(\theta) := F_i(\theta; \widehat{Z}_i)$. We have

$$\mathbb{P}(\mathcal{E}_i^{1,j}) \leq \mathbb{P}\left(F_i(\theta_1) > t\right) + \mathbb{P}\left(F_i(\theta_j) \leq t\right)$$

for all $t \geq 0$. We choose $t = \frac{F^1(\theta_1)+F^1(\theta_j)}{2}$. With this choice, we obtain

$$\mathbb{P}\left(F_i(\theta_1) > t\right) = \mathbb{P}\left(F_i(\theta_1) > \frac{F^1(\theta_1) + F^1(\theta_j)}{2}\right) \tag{37}$$

$$= \mathbb{P}\left(F_i(\theta_1) - F^1(\theta_1) > \frac{F^1(\theta_j) - F^1(\theta_1)}{2}\right). \tag{38}$$

Similarly, for the second term, we have

$$\mathbb{P}\left(F_i(\theta_j) \leq t\right) = \mathbb{P}\left(F_i(\theta_j) - F^1(\theta_j) \leq -\frac{F^1(\theta_j) - F^1(\theta_1)}{2}\right). \tag{39}$$

Based on our assumption, we know that $\|\theta_j - \theta_1\| \geq \Delta - \frac{1}{4}\sqrt{\frac{\lambda}{L}}\Delta \geq \frac{3}{4}\Delta$. According to the strong convexity of $F^1(\cdot)$,

$$F^1(\theta_j) \geq F^1(\theta_1^*) + \frac{\lambda}{2}\|\theta_j - \theta_1^*\|^2 \geq F^1(\theta_1^*) + \frac{9\lambda}{32}\Delta^2,$$

and according to the smoothness of $F^1(\cdot)$,

$$F^1(\theta_1) \leq F^1(\theta_1^*) + \frac{L}{2}\|\theta_1 - \theta_1^*\|^2 \leq F^1(\theta_1^*) + \frac{L}{2}\frac{\lambda}{16L}\Delta^2 = F^1(\theta_1^*) + \frac{\lambda}{32}\Delta^2.$$

Therefore, $F^1(\theta_j) - F^1(\theta_1) \geq \frac{\lambda}{4}\Delta^2$. Then, according to Chebyshev's inequality, we obtain that $\mathbb{P}(F_i(\theta_1) > t) \leq \frac{64\eta^2}{\lambda^2\Delta^4 n'}$ and that $\mathbb{P}(F_i(\theta_j) \leq t) \leq \frac{64\eta^2}{\lambda^2\Delta^4 n'}$, which complete the proof.