Private and Personalized Histogram Estimation in a Federated Setting

Amrith Setlur*
Carnegie Mellon University
asetlur@cs.cmu.edu

Vitaly Feldman
Apple
vitalyf@apple.com

Kunal Talwar
Apple
ktalwar@apple.com

Abstract

Personalized federated learning (PFL) aims at learning personalized models for users in a federated setup. We focus on the problem of privately estimating histograms (in the KL metric) for each user in the network. Conventionally, for more general problems learning a global model jointly via federated averaging, and then finetuning locally for each user has been a winning strategy. But this can be suboptimal if the user distribution observes diverse subpopulations, as one might expect with user vocabularies. To tackle this, we study an alterative PFL technique: clustering based personalization that first identifies diverse subpopulations when present, enabling users to collaborate more closely with others from the same subpopulation. We motivate our algorithm via a stylized generative process: mixture of Dirichlets, and propose initialization/pre-processing techniques that reduce the iteration complexity of clustering. This enables the application of privacy mechanisms at each step of our iterative procedure, making the algorithm user-level differentially private without severe drop in utility due to added noise. Finally, we present empirical results on Reddit users data where we compare our method with other well-known PFL approaches applied to private histogram estimation.

1 Introduction

In many modern, data-intensive applications like recommendation systems, image recognition, and conversational AI, federated learning (FL) has become a vital component to learn from user data that is stored on mobile phones and personal computers while preserving privacy (Konečnỳ et al., 2016). At the same time, FL presents numerous statistical and computational challenges due to its highly decentralized system architecture and heterogeneity in user data distributions (Li et al., 2020). In this work, we focus on tackling the statistical heterogeneity problem through personalized federated learning (PFL) (Wu et al., 2022). Here, the goal is to learn individual predictive models for each user (Fallah et al., 2020). For example, given the same context sentence users would likely differ in their preferences over the next token they may type on their mobile device (Hard et al., 2018). A step towards learning personalized language models (Salemi et al., 2023) would be to first learn models that accurately estimate the marginal token distribution for each user given a few tokens from the user — which is the focus of this work.

We take a closer look at the specific PFL problem of privacy preserving personalized histogram estimation in a federated setting. If each user had infinite samples, then this problem can be solved locally and privately by using only user data. But in practice, user data is limited, and users can potentially benefit significantly from collobarating with others. This is especially true if the histograms share a latent structure, for example, if the users have similar distribution over common

*Part of the work was done during an internship at Apple.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

tokens and differ in rare tokens. One way of collaborating would be to share data between users, but this violates privacy. A winning strategy for satisfying these constraints has been to learn a single global model privately from data across all users (FedAvg (Konečnỳ et al., 2016)), and then finetuning these models for each user locally (Cheng et al., 2021, 2023; Collins et al., 2022). For privacy preserving personalization and multi-task learning, some works have also proposed variants of gradient based optimization approaches that learn a single globally shared structure privately (e.g., latent low rank subspace) before locally adapting it for each user (Jain et al., 2021; Hu et al., 2021). But the above strategies may not be optimal for distributions with diverse and sufficiently well concentrated subpopulations (e.g., mixture of Dirichlets), which can be expected specifically for user histogram distributions.

In this work, we begin by introducing the problem of estimating histograms in Kullback-Leibler (KL) divergence for users in a federated network. We follow this with a stylized model that makes distributional assumptions on the data generative process. For this process, we understand the Bayes optimal estimators for each user's histogram, and define analogous estimators for typical methods like FedAvg, and finetuning. Motivated by this, we propose an algorithm that enables a stronger collaboration between similar cohorts of users, i.e., users having similar empirical data distributions. At a high level, our algorithm uses clustering to identify diverse subpopulations of users and learns a small set of diverse models for each subpopulation. These diverse models are further finetuned (personalized) for each user. We also propose initialization and pre-processing techniques that further reduces iteration complexity for clustering. This allows us to use standard privacy mechanisms that preserve user-level differential privacy (DP) (Dwork, 2010) at each iteration. For appropriate privacy parameters the full algorithm also satisfies user-level DP.

Recently, some prior works (Ghosh et al., 2020; Werner et al., 2022; Marfoq et al., 2021) proposed clustering based methods for canonical PFL problems like optimizing a smooth/strongly convex function over different user distributions (which may not be satisfied by KL metric). Our algorithm can be viewed as an addition to these approaches, but specifically tailored for histogram estimation in the KL metric, with the added benefit of ensuring a strong notion of user participation privacy.

We validate both non-private and private versions of our algorithm on the real world data distribution of Reddit users (Caldas et al., 2018). In the non-private setting, we find that our approach yields at least 5% gains over typical baselines: learning a single global histogram, local histograms for each user, and their combination (finetuning). To a good extent, this validates our distributional assumption that motivated our algorithm. In the private setting, we find that our adaptive mean estimation (per iteration) and pre-processing techniques limit any drop in utility caused by noise addition, ensuring the privacy mechanisms do not eat away gains we observed non-privately.

2 Problem Setup

For learning user-specific histograms, one can naively use just user data alone but this is statistically inefficient since in most cases number of data points (tokens) per user is much lesser than size of the user vocabulary. Thus, if there is any structure (e.g., distribution over common tokens) that is shared across users, a more efficient learning algorithm will first learn the shared structure before personalizing it for each user. One way of recovering this structure is to make some assumptions on the distribution of user histograms, which is what we do later in this section. Before that we introduce some common notations, provide a formal introduction of the problem , and enumerate the different goals for any algorithm intended to solve it.

Notations. We will use P to denote a distribution, $\hat{P}$ for an estimate of P, $\mathcal{P}$ for sets and algorithms (clear from context), and $\mathbf{P}$ for random matrices where $\mathbf{P}_{:,i}/\mathbf{P}_{i,:}$ indexes into the i^{th} column/row respectively. For any iterative procedure where the variables evolve with every step, the variable p at t^{th} timestep is denoted with $p^{(t)}$. The i^{th} axis aligned vector is e_i, where $e_i[j] = 1$ if $j = i$, else 0.

Setup. The histogram for user u, denoted as $Q_u \in \Delta^{|\mathcal{V}|-1}$ is a categorical distribution over a discrete vocabulary $\mathcal{V}$, where $d =: |\mathcal{V}|$. Given, m samples from user u, we can estimate its empirical distribution $\widehat{Q}_u$ where $\widehat{Q}_u \sim {}^1\!/_m \cdot \mathrm{Multinomial}(Q_u, m)$. There exists an unknown meta-distribution D over user histograms Q_u, from which n users are sampled IID, and sampling further from the corresponding multinomials gives us the set $\mathcal{S} =: \{\widehat{Q}_u\}_{u=1}^{n}$.

Goals. The goal is to define an algorithm $\mathcal{A}$, that takes as input the set $\mathcal{S}$ and outputs a functional $\mathcal{A}_\mathcal{S}$. This map $\mathcal{A}_\mathcal{S} : \Delta^{|\mathcal{V}|-1} \mapsto \Delta^{|\mathcal{V}|-1}$ maps the empirical distribution $\widehat{Q}_u$ to $\mathcal{A}_\mathcal{S}(\widehat{Q}_u)$ so that the distance to the true distribution Q_u is low in KL divergence: $\mathrm{KL}(Q_u \mid\mid \mathcal{A}_\mathcal{S}(\widehat{Q}_u))$. There are two specific goals that we outline: (1) utility guarantee: $\mathcal{A}$ looks to minimize the expected KL divergence over D: $\mathbb{E}_\mathcal{S}\mathbb{E}_{Q_u}\mathbb{E}_{\widehat{Q}_u|Q_u}\left[\mathrm{KL}(Q_u \mid\mid \mathcal{A}_\mathcal{S}(\widehat{Q}_u))\right]$; and (2) privacy guarantee: $\mathcal{A}$ must be (ε, δ) user-level differentially private (see Definition 2.1).

Definition 2.1 ((ε, δ)-Differential Privacy (DP) (Dwork et al., 2016))**.** *Given $\varepsilon \geq 0$, $\delta \in [0, 1]$ and a neighbouring relation $\sim$, a randomized mechanism $\mathcal{M} : \mathfrak{D} \to \mathcal{Y}$ from the set of datasets to an output space $\mathcal{Y}$ is (ϵ, δ)-differentially private if for all neighboring datasets $D \sim D' \in \mathfrak{D}$, and all events $E \subseteq \mathcal{Y}$,*

$$\Pr[\mathcal{M}(D) \in E] \leq e^\epsilon \cdot \Pr[\mathcal{M}(D') \in E] + \delta, \tag{1}$$

where probabilities are taken over the randomness of $\mathcal{M}$. When $\delta = 0$, we refer to this as pure ε-DP.

Note that in the aforementioned setup, the algorithm $\mathcal{A}_\mathcal{S}$ outputs a personalized (different) histogram estimate $\mathcal{A}_\mathcal{S}(\widehat{Q}_u)$ for each user u. In general, outlining a personalization algorithm that is minimax optimal for a large class of meta-distributions D may not only be challenging, but may also yield overly pessimistic solutions. For example, when each Q_u is a uniform distribution over a uniformly random subset of $\mathcal{V}$, then optimal $\mathcal{A}_\mathcal{S}$ does not benefit from other users' data, where as when Q_u is identical for all users then $\mathcal{A}_\mathcal{S}(\widehat{Q}_u) \mapsto {}^1\!/\!{}_n \sum_{Q \in S} Q$ is minimax optimal. To avoid such issues, we shall now introduce a stylized generative model that makes some plausible assumptions on D.

2.1 Stylized generative model: mixture of Dirichlets

In this subsection, we introduce assumptions on D by outlining a probabilistic graphical model that underpins the sampling of user vocabularies Q_u. The metadistribution D is a mixture of K Dirichlet distributions: $\{\mathrm{Dir}(\alpha P_1), \mathrm{Dir}(\alpha P_2) \ldots, \mathrm{Dir}(\alpha P_K)\}$. The true histogram Q_u for user u is sampled from D in the following manner:

1. Sample the identity of the underlying Dirichlet distribution (cluster): c_u,

$$\mathbb{P}[c_u = c] = w_c, \;\; \text{where,} \;\; w_c > 0, \; \sum_{c \in [K]} w_c = 1, \;\; \min_c w_c = \Omega(1/k).$$

2. Then, sample $Q_u \sim \mathrm{Dir}(\alpha P_{c_u})$.
3. Recall that for users in $\mathcal{S}$, the empirical distribution $\widehat{Q}_u \sim {}^1\!/\!{}_m \cdot \mathrm{Multinomial}(Q_u, m)$.

As a warmup, we will first go through typical federated learning algorithms in this setup and the closed form realizations of the corresponding estimators. Then, we introduce our estimator which involves identifying cluster (underlying Dirichlet) membership for each user.

FedAvg. One of the most common federated learning algorithms is FedAvg (Konečnỳ et al., 2016) which trains a single global model (single histogram Q_{fa}) that does well on all user datapoints in the set of train users:

$$Q_{\mathrm{fa}} \;\; =: \;\; \underset{Q \in \Delta(\mathcal{V})}{\arg\min} \sum_{\widehat{Q}_u \in \mathcal{S}} \mathrm{KL}(\widehat{Q}_u \mid\mid Q) \tag{2}$$

Lemma 2.1 (FedAvg estimate)**.** *The FedAvg model is given by $Q_{\mathrm{fa}} = \frac{1}{n} \sum_{\widehat{Q}_u \in \mathcal{S}} \widehat{Q}_u$.*

Finetuning. Given a a global model, a popular method to personalize the model for each user is to finetune (Collins et al., 2022; Cheng et al., 2023). In our setup, we use the term finetuning Q for user u for any estimator that takes in $Q, \widehat{Q}_u$ and outputs: $\lambda Q + (1 - \lambda)\widehat{Q}_u$ for $\lambda \in [0, 1]$. For example, in this case finetuning the FedAvg model would mean that the output for user u is: $\lambda Q_{\mathrm{fa}} + (1 - \lambda)\widehat{Q}_u$.

Lemma 2.2 (Bayes optimal estimator)**.** *Given P_{c_u}, α the Bayes optimal estimator is: $(\frac{\alpha}{\alpha+m})P_u + (\frac{m}{\alpha+m})\widehat{Q}_u$, i.e., $\mathbb{E}_{Q_u|P_{c_u}} \mathbb{E}_{\widehat{Q}_u|Q_u}\left[\mathrm{KL}(Q_u \mid\mid ((\frac{\alpha}{\alpha+m})P_u + (\frac{m}{\alpha+m})\widehat{Q}_u))\right]$, matches the loss of the optimal algorithm for P_{c_u}.*

Intractability of Bayes optimal estimator. From Lemma 2.2, we can see that the Bayes optimal predictor finetunes the cluster center P_{c_u} for each user. The problem here is that the cluster centers $\{P_1, \ldots, P_K\}$, as well as the memberships c_u are unknown. In general, if we can compute the posterior distribution over $Q_u \mid \widehat{Q}_u, \mathcal{S}$, the mean of this posterior is the optimal predictor we are looking for (see Appendix A). But computing such a posterior is intractable, even if we assume reasonable priors over $\{P_1, \ldots, P_K\}$. To overcome this challenge, we propose the following approximation.

Proposed estimator. Using the training set S, and knowledge of number of clusters K, we can compute the maximum likelihood estimates (MLE) $\widehat{P}_1, \ldots, \widehat{P}_K$ for the cluster centers. Then, given the empirical data $\widehat{Q}_u$ for a test user, we can compute the MLE for the membership variable $\widehat{c_u}$, conditioned on the estimates $\widehat{P}_1, \ldots, \widehat{P}_K$. Note that computing MLE estimates $\widehat{P}_1, \ldots, \widehat{P}_K$ involves solving a non-concave maximization problem for mixture of Dirichlets, even when $\widehat{Q}_u = Q_u$ for the train users. Typically, this is done using Expectation-Maximization algorithms (EM) (Balakrishnan et al., 2017). In our setting, the distribution Q_u for each user belongs to only one cluster c_u. Hence, we model this as a latent variable and given the maximum likelihood estimate $\widehat{c_u}$ for this variable, with $\mathcal{C}_k =: \{u : \widehat{c_u} = k\}$, the MLE estimate for $\widehat{P}_k$ is: $1/|\mathcal{C}_k| \sum_{\widehat{Q}_u \in \mathcal{C}_k} \widehat{Q}_u$. Based on this simple reduction, we can conclude that the solution for the following clustering problem is realized by the MLE estimates.

$$\min_{\widehat{P}_1, \widehat{P}_2, \ldots, \widehat{P}_K} \sum_{u \in \mathcal{S}} \min_{k \in [K]} \ \mathrm{KL}(\widehat{Q}_u \mid\mid \widehat{P}_k) \tag{3}$$

Since $\widehat{P}_k$ is the cluster averaged model for cluster k, we can personalize the cluster model $\widehat{P}_{\widehat{c_u}}$ by finetuning: $\widehat{Q}_u \mapsto \lambda \widehat{P}_{\widehat{c_u}} + (1 - \lambda)\widehat{Q}_u$, where λ is a hyperparameter tuned on a validation set of users. Based on the above principle, in the next section we discuss an iterative algorithm to discover the cluster the empirical distributions in $\mathcal{S}$, and then estimate the cluster centers by averaging the distributions in each cluster.

3 Non-private and Private Algorithms for Histogram Clustering

We are now ready to introduce iterative algorithms for clustering the empirical distributions in $\mathcal{S}$ (objective in Equation 3), and recover the cluster centers. We first present a non-private algorithm, along with a k-means++ (Arthur and Vassilvitskii, 2007) style initialization scheme. Then, we shall discuss the private version of it, with some pre-processing steps that are crucial in reducing the dimensionality of the problem, and the iteration complexity of the clustering procedure.

First, we use Algorithm 1 to give us well-separated initial cluster centers $\widehat{P}_1^{(0)}, \ldots, \widehat{P}_1^{(K)}$. It takes as input the data matrix $\mathbf{S} \in \mathbb{R}^{n \times d}$ (constructed from $\mathcal{S}$ by ordering clients and slotting their data in the matrix), where row u $\mathbf{S}_{u,:} = \widehat{Q}_u \in \Delta(\mathcal{V})$ is the empirical distribution for user u. It also takes as input the number of clusters k, and a scalar temperature $\tau > 0$. The algorithm begins by picking the first cluster uniformly from the set $\mathcal{S}$. Then, it picks the next cluster from a distribution over the set $\mathcal{S}$, which up weights user u if their data $\widehat{Q}_u$ is not close to any of the already picked centers. It proceeds this way until all K initial centers are chosen. The temperature τ scales the smoothness of the distribution at each iteration. When $\tau \to 0$, we pick the center from a uniform distribution (random initialization), and when $\tau \to \infty$, we pick the most underfit user with probability $\to 1$.

Algorithm 1 Cluster Initialization

Require: $\mathbf{S}$ ordered dataset $\in \mathbb{R}^{n \times d}$, number of clusters K, temperature $\tau > 0$.
Ensure: Initial cluster centers $\hat{P}_1^{(0)}, \ldots, \hat{P}_K^{(0)}$.

1: Sample $j \sim \mathrm{Uniform}(\{1, \ldots, n\})$ and set $\widehat{P}_1^{(0)} \leftarrow \mathbf{S}_{j,:}$.
2: Initialize $k \leftarrow 1$.
3: **while** k < K **do**
4: $\quad k \leftarrow k + 1$.
5: $\quad$ Sample j from $\{1, \ldots, n\}$ where $\mathrm{Prob}(j = u) \propto \exp\left(\tau \cdot \min_{j' \in [k]} \ \mathrm{KL}(\widehat{P}_{j'}^{(0)} \mid\mid \widehat{Q}_u)\right)$.
6: $\quad \widehat{P}_k^{(0)} \leftarrow \mathbf{S}_{j,:}$.
7: **end while**
8: Return $\hat{P}_1^{(0)}, \ldots, \hat{P}_K^{(0)}$.

Next, we use Algorithm 2 to run multiple rounds of the following two steps: (1) re-centering of clusters (step 5), and (2) re-assignement of user distributions to clusters (step 7). Each step greedily reduces the objective value in Equation equation 3. In addition to $\mathbf{S}, k$ it also takes as input the maximum number of clustering iterations. Note that this algorithm is similar to LLoyd's k-means (Ostrovsky et al., 2013), except that the re-assignment step uses the KL metric, as opposed to the euclidean norm for k-means. Directly applying results from Balakrishnan et al. (2017) we can see that under some initialization conditions the rate of mis-clustering (incorrect cluster assignments) goes down exponentially with each iteration of Algorithm 2. As the cluster assignments improve, so does each individual estimate $\widehat{P}_k$ which is given by the average of $\widehat{Q}_u$s assigned to cluster k.

Algorithm 2 Non-private Histogram Clustering

Require: $\mathbf{S}$ ordered dataset $\in \mathbb{R}^{n \times d}$, k number of cluster centers, T maximum iterations
Ensure: Assignment vectors $\hat{\mathbf{C}} \in \mathbb{R}^{n \times k}$ where $\mathbf{C}_{i,:} \in \{e_1, e_2, \ldots, e_k\}$, cluster centers $\widehat{P}_1, \ldots, \widehat{P}_K$

1: Initialize cluster centers $\widehat{P}_1^{(0)}, \ldots, \widehat{P}_k^{(0)}$ using Algorithm 1.
2: Initialize $t \leftarrow 0$.
3: Initialize $\widehat{\mathbf{C}}_{u,:}^{(0)} = e_j$ if $j \in \arg\min_{k \in [K]} \ \mathrm{KL}(\widehat{Q}_u \ || \ \widehat{P}_k^{(0)})$.
4: **while** $t < T$ **do**
5: $\quad \widehat{P}_k^{t+1} \leftarrow \frac{\mathbf{S}^\top \widehat{\mathbf{C}}_{:,k}^{(t)}}{\|\widehat{\mathbf{C}}_{:,k}^{(t)}\|_1}$ if $\|\widehat{\mathbf{C}}_{:,k}^{(t)}\|_1 > 0$, else $[1/d, \ldots, 1/d]^\top$.
6: $\quad t \leftarrow t + 1$
7: $\quad \widehat{\mathbf{C}}_{u,:}^{(t+1)} \leftarrow e_j$ if $j \in \arg\min_{k \in [K]} \ \mathrm{KL}(\widehat{Q}_u \ || \ \widehat{P}_k^{(t+1)})$.
8: **end while**
9: $\widehat{\mathbf{C}} \leftarrow \widehat{\mathbf{C}}^{(t)}, \quad \widehat{P}_k \leftarrow \widehat{P}_k^{(T)} \ \forall k \in [K]$.
10: Return $\widehat{\mathbf{C}}, \widehat{P}_1, \widehat{P}_2, \ldots, \widehat{P}_K$.

In order to make each iteration (step 5-step 7) of Algorithm 2 private, we only need to make the re-centering (step 5) differentially private, since the cluster assignments are computed locally for each user from private estimates $\widehat{P}_1^{(t)}, \ldots, \widehat{P}_K^{(t)}$. Note, that the re-centering step for a cluster simply computes the mean of user distributions that were assigned to it in the previous step. We use adaptive clipping techniques (Andrew et al., 2021; Cummings et al., 2022) to make the mean estimation user-level private. We outline this in Algorithm 3. Essentially, this involves four parts: (1) privately estimate the mean of all datapoints up to some confidence interval determined by the privacy parameters, and statistical hardness (b_1); and (2) clip each vector in $\mathbf{S}$ after subtracting the estimated mean b_1, and scaling appropriately with $\sqrt{b_1}$; (3) average the clipped quantities (by multiplying with $\widehat{C}_{:,k}^{(t)}$) and add much smaller level of privacy noise to get b_2; and finally (4) rescale b_2 using b_1. The key idea behind this procedure is that first step already gives us a reasonable range for the mean. Then, we refine the confidence interval around this mean if users in cluster k have well concentrated distributions, i.e., element wise $Q_u \in \left[b_1 - c\sqrt{b_1/d}, b_1 + c\sqrt{b_1/d}\right]$ with high probability. In Theorem 3.1 we provide formal privacy guarantees for the full algorithm.

Algorithm 3 Private Centering

Require: $\mathbf{S}$ ordered dataset $\in \mathbb{R}^{n \times d}$, $\widehat{\mathbf{C}}^{(t)}$ old cluster assignments, (ε, δ) privacy parameters, smoothing factor s, clipping threshold c for $\mathrm{Clip}_c(x) \mapsto \max(-c, \min(x, c))$.
Ensure: New cluster centers $\widehat{P}_1^{(t+1)}, \ldots, \widehat{P}_K^{(t+1)}$.

1: $\sigma \leftarrow \sqrt{2\log(1.25/\delta)}/\varepsilon$
2: **for** $k = 1$ to K **do**
3: $\quad a \leftarrow \max\left(\|\mathbf{C}_{:,k}\|_1 + \mathrm{Lap}\left(\frac{1}{\varepsilon}\right), 1\right)$
4: $\quad b_1 \leftarrow \max\left(\frac{\mathbf{S}^\top \widehat{\mathbf{C}}_{:,k}^{(t)}}{a_1} + \mathcal{N}(0,\ \sigma^2),\ s\right)$
5: $\quad b_2 \leftarrow \left(\mathrm{Clip}_{c/\sqrt{d}}\left(\mathrm{Diag}(b_1)^{-1/2}(\mathbf{S}^\top - b_1)\right)\widehat{\mathbf{C}}_{:,k}^{(t)}\right) + \mathcal{N}(0,\ c^2\sigma^2))$
6: $\quad \widehat{P}_k^{(t+1)} \leftarrow b_1 + \mathrm{Diag}(b_1)^{1/2}(b_2/a)$
7: **end for**
8: Return $\widehat{P}_1^{(t+1)}, \ldots, \widehat{P}_K^{(t+1)}$.

Theorem 3.1 (privacy guarantee)**.** *For $1 > \varepsilon, \delta > 0$ Algorithm 3 is $(3\varepsilon, 2\delta)$ user-level DP. Consequently, when Algorithm 2 uses Algorithm 3 for the re-centering (step 5), it is (ε', δ') user-level DP where $\varepsilon' = \left(\sqrt{2T\log(1/\delta)} + T^{e^{3\varepsilon}-1}/e^{3\varepsilon}+1\right)3\varepsilon$, $\delta' = (2T+1)\delta$.*

Dimensionality reduction. While the Gaussian and Laplace noise addition mechanisms ensure DP, it may introduce a lot more noise than the tolerance of the clustering algorithm (which scales with the amount of cluster separation). One way of reducing this noise, is to project data onto some low dimensional ($\ll d$) subspace using projection Π, such that the cluster separation in KL metric is rougly preserved, i.e., for any $i \neq j$, $\mathrm{KL}(\Pi(P_i) \mid\mid \Pi(P_j)) \approx \mathrm{KL}(P_i \mid\mid P_j)$. One such subspace is the one spanned by the K cluster centers under our generative assumptions. We identify this subspace by recovering the top k singular vectors for: $\frac{1}{n}\sum_{u\in\mathcal{S}} \hat{V}_u \hat{V}_u^\top$ where $\hat{V}_u =: \mathrm{Diag}(Q_{\mathrm{fa}})^{-1/2}(\widehat{Q}_u - Q_{\mathrm{fa}})$. These vectors capture directions along which clusters are separated. Given the singular vectors $\{v_j\}_{j=1}^k$ we eliminate all tokens x such that $|v_x|$ is smaller than a threshold for any v, because these tokens would not be useful in identifying any cluster. For more discussion see Appendix C.

4 Experimental results on Reddit data

In this section, we compare the non-private and private versions of our algorithm with other PFL baselines on Reddit data that has a vocabulary of 10^3 tokens. We only consider a subset of the Reddit user data with roughly $n = 10^5$ users. We partition the data from each user into train and test sets where the train set uses $m = 5 \times 10^2$ data points (for $\widehat{Q}_u$) and the test set uses 2×10^3 data points (for true Q_u). Additionally, for a fraction of users (5×10^3 from the original set), we also have 10^3 data points for validation. This is used for tuning the hyperparameters like number of clusters k, clipping threshold c, temperature τ, etc.

Method	Avg. KL for non-private version	Avg. KL for private version
Local	5.093	5.093
FedAvg	1.054	1.115
IFCA Ghosh et al. (2020)	1.036	1.110
Algorithm 2 + DimRed	0.971	**0.990**
Algorithm 2	**0.930**	1.053
FedAvg + FT	0.912	0.958
IFCA + FT	0.875	0.948
Algorithm 2 + DimRed + FT	0.883	**0.904**
Algorithm 2 + FT	**0.868**	0.951

Table 1: **Average test KL divergence for non-private and private methods on Reddit data:** Each number is averaged over 20 random runs, and the 95% confidence interval for each value is ± 0.01. For the private versions of the corresponding methods we set $\epsilon = 15, \delta = 10^{-10}$. Following hyperparameters were tuned on the validation set: $k = 10, T = 50, \lambda = 0.3, \tau = 0.5$.

From Table 4 we see that non-privately there is a benefit from assuming diverse subpopulations in the data distribution D. This is validated by the error reduction in KL divergence when comparing Algorithm 2 with FedAvg/Local models (both with and without finetuning (FT)). Note that while IFCA (Ghosh et al., 2020) also makes a similar assumption, our algorithms differ in the per-iteration update. While IFCA only takes a gradient step towards the optimal cluster center, we optimize for the optimal cluster center completely by taking the average (which is the minimizer by virtue of KL). Consequently, in as little as 50 iterations Algorithm 2 converges, but IFCA fails to recover cluster centers with very few iterations. For the performance of private algorithms, we first notice that each private algorithm performs worse than its non-private counterpart. This is expected because of noise injection by privacy mechanisms. Particularly, we note that the gains from Algorithm 2 are still retained when reducing the dimensionality of the empirical user distributions (DimRed). While DimRed slightly hurts performance over using the full set of tokens non-privately, the dimensionality reduction proves to be important in making the clustering procedure somewhat resilient to the noise added by privacy mechanisms.

References

Andrew, G., Thakkar, O., McMahan, B., and Ramaswamy, S. (2021). Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466.

Arthur, D. and Vassilvitskii, S. (2007). K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

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

Balle, B. and Wang, Y.-X. (2018). Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR.

Banerjee, A., Merugu, S., Dhillon, I. S., Ghosh, J., and Lafferty, J. (2005). Clustering with bregman divergences. *Journal of machine learning research*, 6(10).

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Cheng, G., Chadha, K., and Duchi, J. (2021). Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 3.

Cheng, G., Chadha, K., and Duchi, J. (2023). Federated asymptotics: a model to compare federated learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 10650–10689. PMLR.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. (2022). Fedavg with fine tuning: Local updates lead to representation learning. *Advances in Neural Information Processing Systems*, 35:10572–10586.

Cummings, R., Feldman, V., McMillan, A., and Talwar, K. (2022). Mean estimation with user-level privacy under data heterogeneity. *Advances in Neural Information Processing Systems*, 35:29139–29151.

Dwork, C. (2010). Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 174–183. SIAM.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2016). Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Dwork, C., Rothblum, G. N., and Vadhan, S. (2010). Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2020). Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. (2020). An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597.

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

Hu, S., Wu, Z. S., and Smith, V. (2021). Private multi-task learning: Formulation and applications to federated learning. *arXiv preprint arXiv:2108.12978*.

Jain, P., Rush, J., Smith, A., Song, S., and Guha Thakurta, A. (2021). Differentially private model personalization. *Advances in Neural Information Processing Systems*, 34:29723–29735.

Kairouz, P., Oh, S., and Viswanath, P. (2015). The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

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

Marfoq, O., Neglia, G., Bellet, A., Kameni, L., and Vidal, R. (2021). Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34:15434–15447.

Ostrovsky, R., Rabani, Y., Schulman, L. J., and Swamy, C. (2013). The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):1–22.

Salemi, A., Mysore, S., Bendersky, M., and Zamani, H. (2023). Lamp: When large language models meet personalization. *arXiv preprint arXiv:2304.11406*.

Werner, M., He, L., Karimireddy, S. P., Jordan, M., and Jaggi, M. (2022). Towards provably personalized federated learning via threshold-clustering of similar clients. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

Wu, S., Li, T., Charles, Z., Xiao, Y., Liu, Z., Xu, Z., and Smith, V. (2022). Motley: Benchmarking heterogeneity and personalization in federated learning. *arXiv preprint arXiv:2206.09262*.

Appendix

A Missing proofs from Section 2.1

Lemma A.1 (FedAvg estimate). *The FedAvg model is given by* $Q_{\mathrm{fa}} = \frac{1}{n}\sum_{\widehat{Q}_u \in \mathcal{S}} \widehat{Q}_u$.

Proof. From Proposition 1 in Banerjee et al. (2005), we know that $\min_Q (1-\lambda)\,\mathrm{KL}(P_0 \,||\, Q) + \lambda\,\mathrm{KL}(\widehat{Q}_u \,||\, Q) = \lambda P_0 + (1-\lambda)\widehat{Q}_u$. We now apply this for the sets $\mathcal{S}_u = \{\widehat{Q}_u\}$ and set $\mathcal{S}_{-u} = \{\widehat{Q}_i : i \neq q\}$. Doing this recursively for every u gives us the final result. □

Lemma A.2 (Bayes optimal estimator). *Given* P_{c_u}, α *the Bayes optimal estimator is:* $(\frac{\alpha}{\alpha+m})P_u + (\frac{m}{\alpha+m})\widehat{Q}_u$, *i.e.,* $\mathbb{E}_{Q_u | P_{c_u}}\, \mathbb{E}_{\widehat{Q}_u | Q_u}\left[\mathrm{KL}(Q_u \,||\, ((\frac{\alpha}{\alpha+m})P_u + (\frac{m}{\alpha+m})\widehat{Q}_u))\right]$, *matches the loss of the optimal algorithm for* P_{c_u}.

Proof. We note that the distribution of $\widehat{Q}_u$ given the center P_{c_u} is a Dirichlet-Multinomial distribution. Further, the Dirichlet distribution is a conjugate prior for the Multinomial. Hence, the posterior distribution for $Q_u \mid \widehat{Q}_u, P_{c_u}$ is a Dirichlet Multinomial with mean: $(\frac{\alpha}{\alpha+m})P_u + (\frac{m}{\alpha+m})\widehat{Q}_u$. To see why, mean is the Bayes optimal estimator, we once again invoke Proposition 1 from Banerjee et al. (2005). □

B Proof of Theorem 3.1

This proof simply applies known results for the privacy loss of Gaussian and Laplace mechanisms, and then composes the losses with advanced composition.

Privacy for a_1: Since, the noise for the count is sampled from $\mathrm{Lap}(1/\epsilon)$, and the counting query has global sensitivity 1, a_1 is (ε)-DP.

Privacy for b_1: The global sensitivity of the summation $\mathbf{S}^\top \widehat{\mathbf{C}}^{(t)}_{:,k}$ is bounded in ℓ_2 norm by 1, since all probability vectors in $\mathbf{S}$ have ℓ_2 norm of exactly 1. By adding zero mean Gaussian noise with $\sigma = \sqrt{2\log(1.25/\delta)}$ we ensure b_1 satisfies $(2\varepsilon, \delta)$-DP via the privacy loss of Gaussian mechanism (Theorem 1 from Balle and Wang (2018)), followed by basic composition.

Privacy for b_2: Finally for b_2 since we clip at $\pm c/\sqrt{d}$, the global ℓ_2 sensitivity is bounded by c. Thus, adding zero mean Gaussian noise with $\sigma = \sqrt{2c\log(1.25/\delta)}$ suffices for $(\varepsilon, \delta) - DP$ (Theorem 1 from Balle and Wang (2018)). Once, again we can apply basic composition to conclude Algorithm 3 is $(3\varepsilon, 2\delta)$ user-level differentially private.

For privacy analysis of Algorithm 2 with step 5 replaced by Algorithm 3, we can do privacy analysis using the following composition theorem.

Advanced composition (Dwork et al., 2010; Kairouz et al., 2015; Dwork et al., 2014) states that, for any $\delta' > 0$, the *composed* sequence of algorithms is (ε, δ)-differentially private, where

$$\varepsilon = \sqrt{2\log\left(\frac{1}{\delta'}\right)\sum_{m\leq n}\varepsilon_m^2} + \sum_{m\leq n}\varepsilon_m\left(\frac{e^{\varepsilon_m}-1}{e^{\varepsilon_m}+1}\right), \qquad \delta = \delta' + \sum_{m\leq n}\delta_m. \tag{4}$$

When all privacy parameters are the same and small, we roughly have $\epsilon = O(\sqrt{n}\varepsilon_m)$. This means analysts can make extended use of sensitive datasets with a slow degradation of privacy.

Direct application of the above result on the privacy guarantee for Algorithm 3 gives us the final result.

C Dimensionality Reduction

Recall that Q_u is sampled from a mixture of Dirichlets. For simplicity assume $w_1 = w_2 = \ldots = w_K = 1/K$ Let:

$$V_m = \mathrm{Diag}(\bar{Q})^{-1/2}(Q_m - \bar{Q})$$

Here, $\bar{Q} = \frac{1}{K}\sum_{k \in K} P_k$ is the average of cluster centers. Using the second moment of Dirichlet distributions we can do the following derivation:

$$\begin{aligned}
\mathbb{E}[V_m V_m^\top] &= \frac{1}{K}\mathrm{Diag}(\bar{Q})^{-1/2} \sum_{k \in K} \left(\frac{-P_k P_k^\top}{\alpha+1} + P_k P_k^\top + \frac{1}{\alpha+1}\mathrm{Diag}(P_k) \right) \\
&= \frac{1}{K(\alpha+1)} \sum_{k \in [K]} \alpha \mathrm{Diag}(\bar{Q})^{-1/2} P_k P_k^\top \mathrm{Diag}(\bar{Q})^{-1/2}
\end{aligned}$$

Thus, the expected second raw moment of the transformed vectors V_m are given by a sum of K rank one matrices, precisely the matrices defined by outerproducts for the scaled versions of cluster centers. Hence, the top k singular vectors of $\mathbb{E}[V_m V_m^\top]$ would extract a subspace that contains fully the scaled cluster centers, and would retain the cluster separation in ℓ_2^2 distance. Since, we do not have access to $\bar{Q}$, we use the empirical estimate Q_{fa}, and replace the expectation with the plug in estimate as well.

D Analyzing covergence rate for Algorithm 2

The following section outlines some preliminary analysis of Algorithm 2, extending results from Balakrishnan et al. (2017) to our setting. We defer from outlining the results from this analysis in the main paper as it is still an active direction that we are probing for a future version of this draft.

Let us begin by introducing some notaion. For any $S \subseteq [n]$, define $W_S = \sum_{i \in S} w_i$. Recall that $T_g^* = \{i \in [n], z_i = g\}$ and $T_g^{(s)} = \left\{ i \in [n], \hat{z}_i^{(s)} = g \right\}$, let us define

$$S_{gh}^{(s)} = \left\{ i \in [n], z_i = g, \hat{z}_i^{(s)} = h \right\} = T_g^* \cap T_h^{(s)}.$$

Then we have $n_h^{(s)} = \sum_{g \in [k]} n_{gh}^{(s)}$ and $n_h^* = \sum_{g \in [k]} n_{hg}^{(s)}$. In the rest of the proof, we will sometimes drop the upper index (s) of $n_{gh}^{(s)}$, $n_h^{(s)}$ and $S_{gh}^{(s)}$ when there is no ambiguity. Also, we suppress the dependence of k by writing r_k as r. We closely follow the analysis from Balakrishnan et al. (2017), and apply it to the LLoyd's algorithm.

We also assume the following initialization condition:

$$G_0 < \left(\frac{1}{2} - \frac{6}{\sqrt{r_k}} \right) \frac{1}{\lambda} \quad \text{or} \quad \Lambda_0 \le \frac{1}{2} - \frac{4}{\sqrt{r_k}}, \tag{5}$$

Lemma D.1.

$$\|W_S\| \le \sigma \sqrt{3(n+d)|S|} \quad \text{for all } S \subseteq [n]. \tag{6}$$

with probability greater than $1 - \exp(-0.3n)$.

Lemma D.2.

$$\lambda_{max} \left(\sum_{i=1}^{n} w_i w_i' \right) \le 6\sigma^2(n+d). \tag{7}$$

with probability greater than $1 - \exp(-0.5n)$.

Lemma D.3. *For any fixed* $i \in [n]$, $S \subseteq [n]$, $t > 0$ *and* $\delta > 0$,

$$\mathbb{P}\left\{ \left\langle w_i, \frac{1}{|S|}\sum_{j \in S} w_j \right\rangle \ge \frac{3\sigma^2(t\sqrt{|S|} + d + \log(1/\delta))}{|S|} \right\} \le \exp\left(-\min\left\{ \frac{t^2}{4d}, \frac{t}{4} \right\} \right) + \delta.$$

Lemma D.4.

$$\|W_{T_h^*}\| \leq 3\sigma\sqrt{(d+\log n)|T_h^*|} \quad \text{for all } h\in[k] \tag{8}$$

with probability greater than $1-n^{-3}$.

Lemma D.5. *For any fixed* $\theta_1,\cdots,\theta_k\in\mathbb{R}^d$ *and* $a>0$*, we have*

$$\sum_{i\in T_g^*}\mathbb{I}\left\{a\|\theta_h-\theta_g\|^2\leq\langle w_i,\|\theta_h-\theta_g\|\rangle\right\}\leq n_g^*\exp\left(-\frac{a^2\Delta^2}{2\sigma^2}\right)+\sqrt{5n_g^*\log n} \tag{9}$$

for all $g\neq h\in[k]^2$ *with probability greater than* $1-n^{-3}$.

The following two lemmas give the iterative relationship between the error of estimating centers and the error of estimating labels. Let $\mathcal{E}$ be the intersection of high probability events in Lemma D.1, Lemma D.2 Lemma D.4, Lemma D.5 and the initialization condition (5). Then we have $\mathbb{P}\{\mathcal{E}^c\}\leq 3n^{-3}+\nu$. In the rest part of the proof, if not otherwise stated, we all condition on the event $\mathcal{E}$ and the following analysis are deterministic.

Lemma D.6. *On event* $\mathcal{E}$*, if* $G_s\leq\frac{1}{2}$*, then we have*

$$\Lambda_s\leq\frac{3}{r}+\min\left\{\frac{3}{r}\sqrt{kG_s}+2G_s\Lambda_{s-1},\lambda G_s\right\}. \tag{10}$$

Lemma D.7. *On event* $\mathcal{E}$*, if* $\Lambda_s\leq\frac{1-\epsilon}{2}$ *and* $r\geq 36\epsilon^{-2}$*, then*

$$G_{s+1}\leq\frac{2}{\epsilon^4r^2}+\left(\frac{28}{\epsilon^2r}\Lambda_s\right)^2+\sqrt{\frac{5k\log n}{\alpha^2n}}. \tag{11}$$

Proof of Lemma D.6. For any $B\subseteq[n]$, define $\bar{Y}_B=\frac{1}{|B|}\sum_{i\in B}y_i$. The error of estimated centers at step s can be written as

$$\begin{aligned}\hat{\theta}_h^{(s)}-\theta_h &= \frac{1}{n_h}\sum_{i\in S_{hh}}(y_i-\theta_h)+\frac{1}{n_h}\sum_{a\neq h}\sum_{i\in S_{ah}}(y_i-\theta_h)\\ &= \frac{1}{n_h}\sum_{i\in S_{hh}}w_i+\sum_{a\neq h}\frac{n_{ah}}{n_h}\left(\bar{Y}_{S_{ah}}-\theta_h\right)\end{aligned}$$

According to our label update step, we have $\|y_i-\hat{\theta}_h^{(s-1)}\|\leq\|y_i-\hat{\theta}_a^{(s-1)}\|$ for any $i\in S_{ah}$. This means for any $i\in S_{ah}$, y_i is closer to $\hat{\theta}_h^{(s-1)}$ than $\hat{\theta}_a^{(s-1)}$, so is the average of $\{y_i,i\in S_{ah}\}$. Thus, we have

$$\|\bar{Y}_{S_{ah}}-\hat{\theta}_h^{(s-1)}\|\leq\|\bar{Y}_{S_{ah}}-\hat{\theta}_a^{(s-1)}\|.$$

Consequently, triangle inequality gives us

$$\left\|\bar{Y}_{S_{ah}}-\theta_h\right\|\leq\left\|\bar{Y}_{S_{ah}}-\theta_a\right\|+\|\hat{\theta}_a^{(s-1)}-\theta_a\|+\|\hat{\theta}_h^{(s-1)}-\theta_h\|,$$

which, combined with Lemma D.1 and the definition of Λ_{s-1}, yields

$$\left\|\bar{Y}_{S_{ah}}-\theta_h\right\|\leq\sigma\sqrt{3(n+d)/n_{ah}}+2\Lambda_{s-1}\Delta.$$

Taking a weighted sum over $a\neq h\in[k]$, we get

$$\begin{aligned}\sum_{a\neq h}\frac{n_{ah}}{n_h}\left\|\bar{Y}_{S_{ah}}-\theta_h\right\| &\leq \frac{\sigma\sqrt{3(n+d)}}{n_h}\sum_{a\neq h}\sqrt{n_{ah}}+2\Lambda_{s-1}\Delta\sum_{a\neq h}\frac{n_{ah}}{n_h}\\ &\leq \frac{\sigma\sqrt{3(n+d)}}{\sqrt{n_h}}\sqrt{(k-1)G_s}+2G_s\Lambda_{s-1}\Delta,\end{aligned}$$

where the Last inequality is due to Cauchy-Schwartz and the fact that $\sum_{a\neq h}n_{ah}\leq G_sn_h$. Note that $W_{S_{hh}}=W_{T_h^*}-\sum_{a\neq h}W_{S_{ha}}$. Triangle inequality and Lemma D.5 imply

$$\|W_{S_{hh}}\| \leq 3\sigma\sqrt{d+\log n}\sqrt{n_h^*}+\sigma\sqrt{3(n+d)}\sqrt{n_h^*-n_{hh}}.$$

Since $G_s \le \frac{1}{2}$, we have

$$n_h \ge n_{hh} \ge n_h^*(1-G_s) \ge \frac{1}{2}n_h^* \ge \frac{1}{2}\alpha n. \tag{12}$$

Combining the pieces, we obtain

$$\begin{aligned}\left\|\hat{\theta}_h^{(s)} - \theta_h\right\| &\le 3\sigma\sqrt{\frac{d+\log n}{\alpha n}} + 3\sigma\sqrt{\frac{k(n+d)}{\alpha n}G_s} + 2G_s\Lambda_{s-1}\Delta \\ &\le \left(\frac{3}{r}(1+\sqrt{kG_s}) + 2G_s\Lambda_{s-1}\right)\Delta. \end{aligned}\tag{13}$$

Therefore, we get the first term in (10). To prove the second term, we decompose $\hat{\theta}_h^{(s)}$ differently.

$$\begin{aligned}\hat{\theta}_h^{(s)} &= \frac{1}{n_h}\sum_{i=1}^n (\theta_{z_i} + w_i)\,\mathbb{I}\left\{\hat{z}_i^{(s)} = h\right\} \\ &= \frac{1}{n_h}\sum_{a=1}^k\sum_{i=1}^n \theta_a \mathbb{I}\left\{z_i = a, \hat{z}_i^{(s)} = h\right\} + \frac{1}{n_h}\sum_{i\in T_h} w_i \\ &= \sum_{a=1}^k \frac{n_{ah}}{n_h}\theta_a + \frac{1}{n_h}W_{T_h}. \end{aligned}\tag{14}$$

Then, the error of $\hat{\theta}_h^{(s)}$ can be upper bounded as

$$\left\|\hat{\theta}_h^{(s)} - \theta_h\right\| = \left\|\sum_{a=1}^k \frac{n_{ah}}{n_h}(\theta_a - \theta_h) + \frac{1}{n_h}W_{T_h}\right\| \le \left\|\sum_{a\ne h}\frac{n_{ah}}{n_h}(\theta_a - \theta_h)\right\| + \left\|\frac{1}{n_h}W_{T_h}\right\|.$$

By triangle inequality,

$$\left\|\sum_{a\ne h}\frac{n_{ah}}{n_h}(\theta_a-\theta_h)\right\| \le \sum_{a\ne h}\frac{n_{ah}}{n_h}\|\theta_a - \theta_h\| \le \lambda\Delta\sum_{a\ne h}\frac{n_{ah}}{n_h} \le \lambda\Delta G_s. \tag{15}$$

This, together with Lemma D.1 and (12), implies

$$\left\|\hat{\theta}_h^{(s)} - \theta_h\right\| \le \lambda\Delta G_s + \sigma\sqrt{\frac{3(n+d)}{n_h}} \le \left(\lambda G_s + \frac{3}{r}\right)\Delta \tag{16}$$

for all $h \in [k]$. The proof is complete.

□

Proof of Lemma D.7. For any $g \ne h \in [k]\times[k]$,

$$\begin{aligned}\mathbb{I}\left\{z_i = g, \hat{z}_i^{(s+1)} = h\right\} &\le \mathbb{I}\left\{\|\theta_g + w_i - \hat{\theta}_h^{(s)}\|^2 \le \|\theta_g + w_i - \hat{\theta}_g^{(s)}\|^2\right\} \\ &= \mathbb{I}\left\{\|\theta_g - \hat{\theta}_h^{(s)}\|^2 - \|\theta_g - \hat{\theta}_g^{(s)}\|^2 \le 2\left\langle w_i, \hat{\theta}_h^{(s)} - \hat{\theta}_g^{(s)}\right\rangle\right\}. \end{aligned}\tag{17}$$

Triangle inequality implies

$$\|\theta_g - \hat{\theta}_h^{(s)}\|^2 \ge \left(\|\theta_g - \theta_h\| - \|\theta_h - \hat{\theta}_h^{(s)}\|\right)^2 \ge (1-\Lambda_s)^2\|\theta_g - \theta_h\|^2.$$

Using the fact that $(1-x)^2 - y^2 \ge (1-x-y)^2$ when $y(1-x-y)\ge 0$, we obtain

$$\|\theta_g - \hat{\theta}_h^{(s)}\|^2 - \|\theta_g - \hat{\theta}_g^{(s)}\|^2 = (1-2\Lambda_s)^2\|\theta_g - \theta_h\|^2 \ge \epsilon^2\|\theta_g - \theta_h\|^2. \tag{18}$$

Denote by $\Delta_h = \hat{\theta}_h^{(s)} - \theta_h$ for $h \in [k]$. Then,

$$\begin{aligned}&\mathbb{I}\left\{z_i = g, \hat{z}_i^{(s+1)} = h\right\} \\ \le\ &\mathbb{I}\left\{\epsilon^2\|\theta_g - \theta_h\|^2 \le 2\langle w_i, \theta_h - \theta_g + \Delta_h - \Delta_g\rangle\right\} \\ \le\ &\mathbb{I}\left\{\frac{\epsilon^2}{2}\|\theta_g - \theta_h\|^2 \le 2\langle w_i, \theta_h - \theta_g\rangle\right\} + \mathbb{I}\left\{\frac{\epsilon^2}{2}\Delta^2 \le 2\langle w_i, \Delta_h - \Delta_g\rangle\right\}.\end{aligned}$$

Taking a sum over $i \in T_g^*$ and using Markov's inequality on the second term, we obtain

$$n_{gh}^{(s+1)} \leq \sum_{i \in T_g^*} \mathbb{I}\left\{\frac{\epsilon^2}{4}\|\theta_g - \theta_h\|^2 \leq \langle w_i, \theta_h - \theta_g\rangle\right\} + \sum_{i \in T_g^*} \frac{16}{\epsilon^4 \Delta^4}\left(w_i'(\Delta_h - \Delta_g)\right)^2 \tag{19}$$

Note that $\mathbb{I}\left\{\frac{\epsilon^2}{4}\|\theta_g - \theta_h\|^2 \leq \langle w_i, \theta_h - \theta_g\rangle\right\}$ are independent Bernoulli random variables. By Lemma D.5, the first term in RHS of (19) can be upper bounded by

$$n_g^* \exp\left(-\frac{\epsilon^4 \Delta^2}{32\sigma^2}\right) + \sqrt{5 n_g^* \log n}. \tag{20}$$

By Lemma D.2, the second term in RHS of (19) can be upper bounded by

$$\sum_{i \in T_g^*} \frac{16}{\epsilon^4 \Delta^4}\left(w_i'(\Delta_h - \Delta_g)\right)^2 \leq \frac{96(n_g^* + d)\sigma^2}{\epsilon^4 \Delta^4}\|\Delta_g - \Delta_h\|^2. \tag{21}$$

Combining (19), (20) and (21) and using the fact that $\|\Delta_g - \Delta_h\|^2 \leq 4\Lambda_s^2 \Delta^2$, we get

$$n_{gh}^{(s+1)} \leq n_g^* \exp\left(-\frac{\epsilon^4 \Delta^2}{32\sigma^2}\right) + \sqrt{5 n_g^* \log n} + \frac{384(n_g^* + d)\sigma^2}{\epsilon^4 \Delta^2}\Lambda_s^2.$$

Consequently,

$$\max_{g \in [k]} \sum_{h \neq g} \frac{n_{gh}^{(s+1)}}{n_g^*} \leq k \exp\left(-\frac{\epsilon^4 \Delta^2}{32\sigma^2}\right) + k\sqrt{\frac{5 \log n}{\alpha n}} + \frac{384}{\epsilon^4 r^2}\Lambda_s^2. \tag{22}$$

Since $\Lambda_s \leq 1/2$ and $r \geq 20\epsilon^{-2}$, the RHS of (22) is smaller that $1/2$ when $\alpha n \geq 32k^2 \log n$. Thus,

$$n_h^{(s+1)} \geq n_{hh}^{(s+1)} \geq \frac{1}{2} n_h^* \geq \frac{1}{2}\alpha n$$

for all $h \in [k]$ and we have

$$\max_{h \in [k]} \sum_{g \neq h} \frac{n_{gh}^{(s+1)}}{n_h^{(s+1)}} \leq \frac{2}{\alpha} \exp\left(-\frac{\epsilon^4 \Delta^2}{32\sigma^2}\right) + \sqrt{\frac{5k \log n}{\alpha^2 n}} + \frac{768}{\epsilon^4 r^2}\Lambda_s^2, \tag{23}$$

which, together with (22), implies

$$G_{s+1} \leq \exp\left(-\frac{\epsilon^4 \Delta^2}{32\sigma^2} + \log(2/\alpha)\right) + \sqrt{\frac{5k \log n}{\alpha^2 n}} + \frac{768}{\epsilon^4 r^2}\Lambda_s^2$$

Under the assumptions that $\epsilon^4 \alpha \Delta^2/\sigma^2 \geq r^2 \epsilon^4 \geq 36$, we have the desired result (11). ☐

Proof. From Lemma D.6, a necessary condition for $\Lambda_0 \leq \frac{1}{2} - \frac{4}{\sqrt{r}}$ is $G_0 \leq (\frac{1}{2} - \frac{6}{\sqrt{r}})\frac{1}{\lambda}$. Setting $\epsilon = \frac{7}{\sqrt{r}}$ in Lemma D.7, we have $G_1 \leq 0.35$. Plugging it into Lemma D.6 gives us $\Lambda_1 \leq 0.4$, under the assumption that $r \geq 16\sqrt{k}$. Then it can be easily proved by induction that $G_s \leq 0.35$ and $\Lambda_s \leq 0.4$ for all $s \geq 1$. Consequently, Lemma D.6 yields

$$\Lambda_s \leq \frac{3}{r} + \frac{3}{r}\sqrt{k G_s} + G_s \leq \frac{1}{2} + G_s$$

which, combined with (11), implies

$$G_{s+1} \leq \frac{C}{r^2} + \frac{C}{r^2}\left(\frac{1}{4} + 2G_s + G_s^2\right) + \sqrt{\frac{5k \log n}{\alpha^2 n}} \leq \frac{2C}{r^2} + \frac{3C}{r^2} G_s + \sqrt{\frac{5k \log n}{\alpha^2 n}}$$

for some constant C. Here we have chosen $\epsilon = 1/5$ in Lemma 11 to get the first inequality. ☐

Proof. From the proof of Lemma D.6, the error of estimating θ_h at iteration s can be written as $\hat{\theta}_h^{(s)} - \theta_h = \frac{1}{n_h} W_{T_h^*} + u_h$, with

$$\|u_h\| \leq \left(\frac{3}{r} \sqrt{k G_s} + G_s \right) \Delta \leq \sqrt{G_s} \Delta \tag{24}$$

In addition, by Lemma D.6 and Lemma D.7, there is a constant C_1 such that

$$\Lambda_s \leq \frac{3}{r} + \sqrt{G_s} + 2 G_s \Lambda_{s-1} \leq \frac{C_1}{r} + \frac{C_1}{r} \Lambda_{s-1} + 0.7 \Lambda_{s-1} + \left(\frac{C_1 k \log n}{\alpha^2 n} \right)^{1/4}$$

for all $s \geq 1$. Therefore, when r is large enough, we have

$$\Lambda \leq C_2 r^{-1} + C_2 \left(\frac{k \log n}{\alpha^2 n} \right)^{1/4}$$

for all $s \geq \log n$. Then by (18), we have

$$\mathbb{I} \left\{ z_i = g, \hat{z}_i^{(s+1)} = h \right\} \leq \mathbb{I} \left\{ \beta_1 \|\theta_g - \theta_h\|^2 \leq 2 \langle w_i, \theta_h - \theta_g + \Delta_h - \Delta_g \rangle \right\}$$

where $(1 - 2\Lambda_s)^2 \geq \beta_1 := 1 - 4 C_2 r^{-1} - 4 C_2 \left(\frac{k \log n}{\alpha^2 n} \right)^{1/4}$.

In order to prove that A_s attains convergence rates, we first upper bound the expectation of A_s and then derive the high probability bound using Markov's inequality. Similar to the two-mixture case, we need to upper bound the inner product $\langle w_i, \Delta_h - \Delta_g \rangle$ more carefully. Note that $\{T_h^*, h \in [k]\}$ are deterministic sets, we could use concentration equalities to upper bound $W_{T_h^*}$ and u_h parts separately.

Let $v_h = \frac{1}{n_h} W_{T_h^*}$ for $h \in [k]$ and we decompose $\mathbb{I} \left\{ z_i = g, \hat{z}_i^{(s+1)} = h \right\}$ into three terms.

$$\begin{aligned} \mathbb{I} \left\{ z_i = g, \hat{z}_i^{(s+1)} = h \right\} \quad \leq \quad & \mathbb{I} \left\{ \beta \|\theta_g - \theta_h\|^2 \leq 2 \langle w_i, \theta_h - \theta_g \rangle \right\} \\ & + \mathbb{I} \left\{ \beta_2 \Delta^2 \leq 2 \langle w_i, u_h - u_g \rangle \right\} \\ & + \mathbb{I} \left\{ \beta_4 \Delta^2 \leq 2 \langle w_i, v_h - v_g \rangle \right\}, \end{aligned}$$

where β_2 and β_4 will be specified later and $\beta = \beta_1 - \beta_2 - \beta_4$. Taking a sum over $h \in [k]$ and $i \in [n]$, we obtain

$$\mathbb{E} A_{s+1} \leq \mathbb{E} J_1 + \mathbb{E} J_2 + \mathbb{E} J_3$$

with

$$J_1 = \sum_{h \in [k]} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \beta \|\theta_{z_i} - \theta_h\|^2 \leq 2 \langle w_i, \theta_h - \theta_{z_i} \rangle \right\} \tag{25}$$

$$J_2 = \sum_{h \in [k]} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \beta_2 \Delta^2 \leq 2 \langle w_i, u_h - u_{z_i} \rangle \right\}. \tag{26}$$

$$J_3 = \sum_{h \in [k]} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I} \left\{ \beta_4 \Delta^2 \leq 2 \langle w_i, v_{z_i} - v_h \rangle \right\}. \tag{27}$$

Let us first consider the expectation of J_1. Using Chernoff's bound, we have

$$\mathbb{P} \left\{ \beta \|\theta_g - \theta_h\|^2 \leq 2 \langle w_i, \theta_h - \theta_g \rangle \right\} \leq \exp \left(- \frac{\beta^2 \|\theta_h - \theta_g\|^2}{8 \sigma^2} \right) \leq \exp \left(- \frac{\beta^2 \Delta^2}{8 \sigma^2} \right).$$

Thus,

$$\mathbb{E} J_1 \leq k \exp \left(- \frac{\beta^2 \Delta^2}{8 \sigma^2} \right) = \exp \left(- \frac{\gamma \Delta^2}{8 \sigma^2} \right),$$

with $\gamma = \beta^2 - \frac{8 \sigma^2 \log k}{\Delta^2} \geq \beta^2 - 8/r^2$.

We use Markov Inequality to upper bound J_2. Markov's inequality and Lemma D.2 give us

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left\{\beta_2\Delta^2 \le 2\left\langle w_i, u_h - u_{z_i}\right\rangle\right\} &\le \frac{4}{n\beta_2^2\Delta^4}\sum_{g\in[k]}\sum_{i\in T_g^*}\left(w_i'(u_h-u_g)\right)^2 \\
&\le \frac{24\sigma^2}{n\beta_2^2\Delta^4}\sum_{g\in[k]}(n_g^*+d)\|u_h-u_g\|^2.
\end{aligned}$$

(24) implies

$$J_2 \le \frac{96\sigma^2 G_s}{n\beta_2^2\Delta^2}\sum_{h\in[k]}\sum_{g\in[k]}(n_g^*+d) \le \frac{96\sigma^2 k(n+kd)}{\alpha n\beta_2^2\Delta^2}A_s = \frac{12\sqrt{k}}{r}A_s.$$

Here the second inequality is due to the fact that $G_s \le A_s/\alpha$. And we choose $\beta_2 = \sqrt{8k/r}$ in the last equality.

Finally, we upper bound the expectation of J_3. Given $z_i = g$, we have

$$\begin{aligned}
&\mathbb{P}\left\{\beta_4\Delta^2 \le 2\left\langle w_i, v_g - v_h\right\rangle\right\} \\
\le\ &\mathbb{P}\left\{\frac{\beta_4}{4}\Delta^2 \le \left\langle w_i, v_g\right\rangle\right\} + \mathbb{P}\left\{-\frac{\beta_4}{4}\Delta^2 \ge \left\langle w_i, v_h\right\rangle\right\} \\
\le\ &\mathbb{P}\left\{\frac{\beta_4}{8}\Delta^2 \le \left\langle w_i, \frac{1}{n_g^*}W_{T_g^*}\right\rangle\right\} + \mathbb{P}\left\{-\frac{\beta_4}{8}\Delta^2 \ge \left\langle w_i, \frac{1}{n_h^*}W_{T_h^*}\right\rangle\right\}
\end{aligned}$$

Choosing $t = \max\{\frac{\sqrt{d}\Delta}{\sigma}, \frac{\Delta^2}{\sigma^2}\}$, $\delta = \exp\left(-\frac{\Delta^2}{4\sigma^2}\right)$ in Lemma D.3, and

$$\beta_4 = \frac{64}{r} \ge \frac{8}{\Delta^2}\left(\frac{3\max\{\sqrt{d}\sigma\Delta, \Delta^2\}}{\sqrt{\alpha n}} + \frac{3\sigma^2 d + \Delta^2}{\alpha n}\right),$$

we obtain $\mathbb{P}\left\{\beta_4\Delta^2 \le 2\left\langle w_i, v_g - v_h\right\rangle\right\} \le 2\exp(-\Delta^2/(4\sigma^2))$, where we have used the assumption that $n_g^* \ge \alpha n$ and $\alpha n \ge 36r^2$. Thus,

$$\mathbb{E}J_3 \le 2k\exp\left(-\frac{\Delta^2}{\sigma^2}\right),$$

Combining the pieces, we have

$$\begin{aligned}
\mathbb{E}A_{s+1} &\le \mathbb{E}\left[J_1\right] + \mathbb{E}\left[J_2\mathbb{I}\{\mathcal{E}\}\right] + \mathbb{E}\left[J_3\right] + \mathbb{P}\{\mathcal{E}^c\} \\
&\le \exp\left(-\frac{\gamma\Delta^2}{8\sigma^2}\right) + \frac{12\sqrt{k}}{r}\mathbb{E}A_s + 2k\exp\left(-\frac{\Delta^2}{\sigma^2}\right),
\end{aligned}$$

with $\gamma = (\beta_1 - \sqrt{8k/r} - 64/r)^2 - 8/r^2 = 1 - o(1)$. Here only prove the case that $r \to \infty$. For the finite case, all the $o(1)$ in the following proof can be substituted by a small constant.

$$\mathbb{E}A_s \le \frac{1}{2^{s-\lceil\log r\rceil}} + 2\exp\left(-(1-\eta)\frac{\Delta^2}{8\sigma^2}\right) + \frac{2}{n^3} \le 2\exp\left(-(1-\eta)\frac{\Delta^2}{8\sigma^2}\right) + \frac{3}{n^3}$$

when $s \ge 4\log n$. By Markov's inequality, for any $t > 0$,

$$\mathbb{P}\left\{A_s \ge t\right\} \le \frac{1}{t}\mathbb{E}A_s \le \frac{2}{t}\exp\left(-(1-\eta)\frac{\Delta^2}{8\sigma^2}\right) + \frac{3}{n^3 t}. \tag{28}$$

If $(1-\eta)\frac{\Delta^2}{8\sigma^2} \le 2\log n$, choose $t = \exp\left(-(1-\eta-\frac{8\sigma}{\Delta})\frac{\Delta^2}{8\sigma^2}\right)$ and we have

$$\mathbb{P}\left\{A_s \ge \exp\left(-(1-\eta-\frac{8\sigma}{\Delta})\frac{\Delta^2}{8\sigma^2}\right)\right\} \le \frac{4}{n} + 2\exp\left(-\frac{\Delta}{\sigma}\right).$$

Otherwise, since A_s only takes discrete values of $\{0, \frac{1}{n}, \cdots, 1\}$, choosing $t = \frac{1}{n}$ in (28) leads to

$$\mathbb{P}\left\{A_s > 0\right\} = \mathbb{P}\left\{A_s \ge \frac{1}{n}\right\} \le 2n\exp(-2\log n) + \frac{3}{n^2} \le \frac{4}{n}.$$

The proof is complete. $\square$