Personalized PCA for Federated Heterogeneous Data

Kaan Ozkara, Bruce Huang and Suhas Diggavi
Department of Electrical and Computer Engineering, UCLA
kaan@g.ucla.edu, brucehuang@g.ucla.edu, suhas@ee.ucla.edu

***Abstract*—As the high dimensional data generation/storage shifts from data centers to millions of edge devices, PCA algorithms also need to adapt to federated systems to reveal insights about the distributed data. One of the prominent challenges in Federated Learning (FL) is that each edge device has a limited number of samples, and therefore collaboration among clients is necessary for learning tasks. Another challenge is heterogeneous distribution of data across devices, which necessitates careful design of algorithms that enable collaboration of devices with different data distributions. While many such federated supervised learning algorithms were proposed in recent years, heterogeneity for unsupervised FL algorithms (such as PCA) has received less attention. In this work, our goal is to enable collaborations of heterogeneous clients in learning personalized Principal Components (PCs). To this end, we develop a hierarchical Bayesian framework for discovering individual PCs; and inspired by this, we formulate an optimization problem related to maximum likelihood estimation of the PCs. To solve the optimization problem, we propose an alternating Stiefel gradient descent algorithm. Analytically, we prove the convergence result for our proposed algorithm; and empirically, we show that our method outperforms local and global estimation of PCs in various heterogeneous settings in terms of the reconstruction error.**

I. Introduction

Principal Component Analysis (PCA) is one of the most commonly studied unsupervised learning algorithms due to its use in dimensionality reduction and feature learning from high-dimensional data. With the increased computational resources and data generation in edge devices, there has been an interest in federated/distributed PCA recently [1]–[5]. However, most of the proposed work in literature (with the exception of [1]) consider a setting where data across devices are generated from the same distribution (homogeneously); accordingly, they try to construct global PCs that works well for all clients, using either one-shot algorithms [2] or multi-round algorithms [3]–[5]. In contrast, [1] considers a heterogeneous setting where edge PCs are modeled through a homogeneous (globally shared) part and completely independent individual (local) parts. As far as we are aware, [1] is the first personalized PCA work in the literature. A downside of [1] is that the method ceases to function when data is not generated via separation of homogeneous and independent individual parts.

Personalization in supervised FL is well-studied [6]–[11]. In particular, [11], [12] introduced a hierarchical/empirical Bayes framework on personalization that unifies many of the previously proposed ideas. The idea is to construct an empirical Bayes MLE problem where the parameters of the global distributions are learned together with the local (personalized) parameters collaboratively. Combining the technique in [11] with the MLE view of PCA [13], we propose a novel personalized PCA algorithm that does not require the separability of homogeneous and independent parts as in [1]. Our contributions are as follows:

- **Statistical Formulation:** As far as we are aware, we are the first to develop a hierarchical Bayes framework for modeling data heterogeneity applied to PCA.
- **Problem:** We formulate an optimization problem based on MLE of PCs in the hierarchical Bayes model.
- **Algorithm:** We propose an alternating Stiefel gradient descent algorithm for our optimization problem.
- **Convergence:** Analytically, we show that the algorithm converges to a stationary point with a rate of $\mathcal{O}(\frac{1}{T})$, where T is the number of iterations. Furthermore, we give insights on the relation between the amount of heterogeneity and convergence speed.
- **Experiments:** Empirically, we show that our proposed algorithm outperforms the local and global estimation of PCs in terms of the reconstruction error.

Outline. In Section II, we formulate our problem and discuss the probabilistic motivation behind it. Then, we state some preliminary mathematical tools that helps us analyze updates on the Stiefel manifold. In Section III, we propose an alternating Stiefel gradient descent algorithm to optimize the formulated problem, and show its convergence properties whose proof outline is given in Section IV (detailed are in [14]). Lastly, in Section V, we compare our method to local PCA and global PCA empirically and discuss our findings.

II. Preliminaries and Problem Formulation

In this section we develop the probabilistic model for personalization and preliminaries required to derive our results.

A. Preliminaries

The Stiefel manifold, $St(d,r)$, is the set of all orthonormal matrices embedded in a $d \times r$ Euclidean space, $St(d,r) \coloneqq \{\boldsymbol{U} \in \mathbb{R}^{d\times r} | \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}\}$. For any point $\boldsymbol{U} \in St(d,r)$, the tangent space at $\boldsymbol{U}$ is defined as $\mathcal{T}_{\boldsymbol{U}} \coloneqq \{\boldsymbol{V} \in \mathbb{R}^{d\times r} | \boldsymbol{V}^\top \boldsymbol{U} + \boldsymbol{U}^\top \boldsymbol{V} = 0\}$. Accordingly, the projection onto the tangent space at point $\boldsymbol{U}$ can be defined as $\mathcal{P}_{\mathcal{T}_{\boldsymbol{U}}}(\boldsymbol{V}) \coloneqq \boldsymbol{V} -$

This work was supported in part by NSF grants 2139304, 2007714 and 1955632, and Army Research Laboratory grant under Cooperative Agreement W911NF-17-2-0196.

$\frac{1}{2}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{V}+\boldsymbol{V}^\top\boldsymbol{U})$. In general, a lifting is a map from the Stiefel manifold to a tangent space at a point on the manifold and $\mathcal{P}_{\mathcal{T}_{\boldsymbol{U}}}$ is a particular lifting also known as the orthographic lifting, a lifting that is not unique and only defined locally. A retraction at a point $\boldsymbol{U} \in St(d,r)$ is a map $\mathcal{R}_{\boldsymbol{U}} : \mathcal{T}_{\boldsymbol{U}} \to St(d,r)$ that induces local coordinates on the Stiefel manifold (see [15] for more details). In this work we use polar retraction that is defined as $\mathcal{R}_{\boldsymbol{U}}(\boldsymbol{V}) = (\boldsymbol{U}+\boldsymbol{V})(\boldsymbol{I}+\boldsymbol{V}^\top\boldsymbol{V})^{-\frac{1}{2}}$. The polar retraction is a second order retraction that approximates the exponential mapping up to second order terms. Consequently, it possesses the following non-expansiveness property.

Lemma 1 (Non-expansiveness of polar retraction [16])**.** *Let $\boldsymbol{V} \in St(d,r)$, for any point $\boldsymbol{U} \in \mathcal{T}_{\boldsymbol{V}}$ with bounded norm, $\|\boldsymbol{U}\|_F \leq M$, there exists $C \in \mathbb{R}$ such that*

$$\|\mathcal{R}_{\boldsymbol{V}}(\boldsymbol{U}) - (\boldsymbol{V}+\boldsymbol{U})\|_F \leq C\|\boldsymbol{U}\|_F^2. \tag{1}$$

B. Problem Formulation and the Probabilistic View

In a client-server configuration, suppose we have m clients and each client has a dataset, $\boldsymbol{Y}_i \in \mathbb{R}^{d\times n}$, containing n samples of d dimensional vectors. The sample covariance matrix of a client is defined as $\boldsymbol{S}_i = \frac{1}{n}\boldsymbol{Y}_i\boldsymbol{Y}_i^\top$. In contrast to the traditional maximum variance view of PCA, the probabilistic view of PCA [13] uses latent random variables $\boldsymbol{x}$ to model a MLE problem that outputs the PCs. For any data point $\boldsymbol{y} \in \mathbb{R}^d$ at client i, we have the following linear equation:

$$\boldsymbol{y} = \boldsymbol{U}_i\boldsymbol{x} + \boldsymbol{\epsilon} \tag{2}$$

where $\boldsymbol{U}_i \in \mathbb{R}^{d\times r}$ is the PC matrix that relates the latent space to the observation space ($r < d$ for dimensionality reduction), and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\epsilon^2\boldsymbol{I})$ for some intrinsic noise value σ_ϵ^2 and in general it is assumed that $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{I})$. Equation (2) induces a distribution on the observations: $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{U}_i\boldsymbol{U}_i^\top + \sigma_\epsilon^2\boldsymbol{I})$. In addition to data generation, we further make a generative assumption on $\boldsymbol{U}_i$'s, that is, $\boldsymbol{U}_i$'s are generated according to some population distribution $\mathbb{P}(\Gamma)$, which is parametrized by a set of global/population parameters Γ. Note that the hierarchical Bayes model is similar to ones in [11], [12], that is, we have the hierarchy chain $\Gamma \to \boldsymbol{U}_i \to \boldsymbol{Y}_i$. Also, note that this model captures [1], where some PCs are the same across clients (globally shared PCs) and others are uniformly sampled (local PCs). Given this probabilistic model, we can create an MLE problem to find the unknown personalized parameters $\{\boldsymbol{U}_i\}_{i=1}^m$ and the global parameter Γ:

$$\underset{\Gamma,\{\boldsymbol{U_i}\}_{i=1}^m}{\arg\max}\ p(\{\boldsymbol{y}_{ij}\}_{i,j}^{m,n}, \{\boldsymbol{U}_i\}_i^m|\Gamma)$$

In this paper, we focus on a particular prior distribution. We assume that $\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i) \sim \mathcal{N}(\boldsymbol{0}, \sigma_U^2\boldsymbol{I})$. Effectively, there is a latent distribution on the tangent space that induces a distribution of $\boldsymbol{U}_i$s on the Stiefel manifold which dictates that $\boldsymbol{U}_i$s are concentrated. As a result, we can extend the MLE problem with the projected random variables:

$$\underset{\boldsymbol{V},\{\boldsymbol{U_i}\}_{i=1}^m}{\arg\max}\ p(\{\boldsymbol{y}_{ij}\}_{i,j}^{m,n}, \{\boldsymbol{U}_i\}_i^m, \{\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)\}_i^m|\boldsymbol{V})$$

$$= \underset{\boldsymbol{V},\{\boldsymbol{U_i}\}_{i=1}^m}{\arg\max} \prod_{i=1}^m\prod_{j=1}^n p(\boldsymbol{y}_{ij}|\boldsymbol{U}_i)\prod_{i=1}^m p(\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)|\boldsymbol{V}) \tag{3}$$

where $\boldsymbol{y}_{ij}$ denotes j'th sample at client i. Note that in the equation, we use the fact that $y_{ij}|\boldsymbol{U}_i$ is independent of $(\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i), \boldsymbol{V})$ and $(\boldsymbol{U}_i, \mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)|\boldsymbol{V}) = (\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)|\boldsymbol{V})$ as the projection being an invertible function in the neighborhood of $\boldsymbol{V}$ makes $\boldsymbol{U}_i|\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i), \boldsymbol{V}$ deterministic. Taking the negative logarithm of (3) and noting that $\boldsymbol{U}_i, \boldsymbol{V}$ are PC matrices, we obtain the following regularized and constrained optimization problem:

$$\underset{\boldsymbol{V},\{\boldsymbol{U_i}\}}{\arg\min}\ \frac{1}{m}\sum_{i=1}^m \frac{n}{2}(\log(|\boldsymbol{W}_i|) + \mathrm{tr}(\boldsymbol{W}_i^{-1}\boldsymbol{S}_i)) + \frac{d^2(\boldsymbol{V},\boldsymbol{U}_i)}{2\sigma_U^2}$$
$$\text{s.t. } \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I},\ \boldsymbol{U}_i^\top\boldsymbol{U}_i = \boldsymbol{I} \quad \forall i \in [m] \tag{4}$$

where $W_i = (\boldsymbol{U}_i\boldsymbol{U}_i^\top + \sigma_\epsilon^2\boldsymbol{I})$, $d^2(\boldsymbol{V},\boldsymbol{U}_i) = \|\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)\|_F^2$, and $[m] \coloneqq \{1,\dots,m\}$. More compactly, we can define $f(\boldsymbol{V},\{\boldsymbol{U}_i\}_{i=1}^m) \coloneqq \frac{1}{m}\sum_{i=1}^m f_i(\boldsymbol{V},\boldsymbol{U}_i)$, where $f_i(\boldsymbol{V},\boldsymbol{U}_i) \coloneqq \frac{n}{2}(\log(|\boldsymbol{W}_i|) + \mathrm{tr}(\boldsymbol{W}_i^{-1}\boldsymbol{S}_i)) + \frac{d^2(\boldsymbol{V},\boldsymbol{U}_i)}{2\sigma_U^2}$. The Bayesian view introduces a regularization in the optimization problem and allows collaborating through the global PC, $\boldsymbol{V}$, while learning personalized PCs, $\{\boldsymbol{U}_i\}$. Note that overall problem is non-convex.

III. Main Results

Algorithm 1 Alternating Stiefel Gradient Descent on the Steifel manifold for optimizing (4)

Input: Number of iterations T, local sample covariance matrices $\{\boldsymbol{S}_i\}_{i=1}^m$, and learning rates (α,β).

Initialize local PCs $\{\boldsymbol{U}_{i,0}\}_{i=1}^m$ and global PC $\boldsymbol{V}_0$.
for $t = 1$ **to** T **do**
  On Clients:
  for $i = 1$ **to** m: **do**
    Receive $\boldsymbol{V}_{t-1}$
    $\boldsymbol{g}_{i,t-1} = P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t-1}))$
    $\boldsymbol{U}_{i,t} \leftarrow \mathcal{R}_{\boldsymbol{U}_{i,t-1}}(-\alpha\boldsymbol{g}_{i,t-1})$
    $\boldsymbol{h}_{i,t-1} = P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t}))$
    $\boldsymbol{V}_{i,t} \leftarrow \boldsymbol{V}_{t-1} - \beta\boldsymbol{h}_{i,t-1}$
    Send $\boldsymbol{V}_{i,t}$ to Server
  end for
  At the Server:
  Receive $\{\boldsymbol{V}_{i,t}\}_{i=1}^m$
  $\boldsymbol{V}_t = \mathcal{R}_{\boldsymbol{V}_{t-1}}(\frac{1}{m}\sum_{i=1}^m \boldsymbol{V}_{i,t} - \boldsymbol{V}_{t-1}) = \mathcal{R}_{\boldsymbol{V}_{t-1}}(-\beta\frac{1}{m}\sum_{i=1}^m P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t})))$
  Broadcast $\boldsymbol{V}_t$
end for

Output: Personalized PCs $\{\boldsymbol{U}_{1,T},\dots,\boldsymbol{U}_{m,T}\}$.

Algorithm. The challenge in optimizing (4) is that we need to enforce the orthogonality constraints so that $\boldsymbol{U}_i$'s and $\boldsymbol{V}$ are valid PCs at each iteration. To that end, we designed Algorithm 1 to optimize (4) with alternating Stiefel gradient descent. We use Stiefel (projected) gradients to make sure that the updates stays on the tangent space with respect to the

manifold. After the gradient updates, we retract the parameters back to Stiefel manifold to keep feasibility.

In particular, each client receives the global PC $\boldsymbol{V}_{t-1}$ in line 5. Firstly, clients compute the projected gradient with respect to personalized PC, $\boldsymbol{U}_i$, and update it using the polar retraction. Lastly, the local versions of global PC, $\boldsymbol{V}_i$'s, are created through projected gradients on the received global PC and are sent to the server.

A. Convergence Result

In this section, we analyze the first order convergence properties of Algorithm 1. Alongside Lemma 1, we need other intermediate results for our analysis:

Lemma 2 (Lipschitz type inequality [16])**.** *Let $\boldsymbol{U}, \boldsymbol{V} \in St(d,r)$. If a function ψ is L-Lipschitz smooth in $\mathbb{R}^{d\times r}$, the following inequality holds:*

$$|\psi(\boldsymbol{V})-(\psi(\boldsymbol{U})+\langle P_{\mathcal{T}_{\boldsymbol{U}}}(\nabla\psi(\boldsymbol{U})),\boldsymbol{V}-\boldsymbol{U}\rangle)| \leq \frac{L_g}{2}\|\boldsymbol{V}-\boldsymbol{U}\|_F^2$$

where $L_g = L + G$ with $G \coloneqq \max_{\boldsymbol{U}\in St(d,r)} \|\nabla\psi(\boldsymbol{U})\|_2$.

Lemma 2 helps us translate Euclidean analysis techniques to Stiefel manifold. Before stating our convergence result, we assume the following property for the sample data covariance matrices $\boldsymbol{S}_i$'s.

Asssumption 1. *For each client i, the operator and Frobenius norms of $\boldsymbol{S}_i$ are bounded by*

$$\|\boldsymbol{S}_i\|_F \leq G_{i,F} \quad \textit{and} \quad \|\boldsymbol{S}_i\|_{op} \leq G_{i,op},$$

and we define $G_{max,F} \coloneqq \max_{i\in[m]} G_{i,F}$ and $G_{max,op} \coloneqq \max_{i\in[m]} G_{i,op}$.

Assumption 1 corresponds to assuming the loss function is Lipschitz smooth with respect to $\boldsymbol{U}_i$ and $\boldsymbol{V}$. Given Assumption 1, we have the following lemmas:

Lemma 3. *$f_i(\boldsymbol{V},\boldsymbol{U}_i)$ is L_U-Lipschitz smooth with respect to $\boldsymbol{U}_i$ and $\|\nabla f_i(\boldsymbol{V},\boldsymbol{U}_i)\|_2 \leq G_U$ for all $i \in [m]$ with constants $L_U := \frac{n}{2}\left(\frac{1}{\sigma_\epsilon^2} + \frac{G_{max,op}}{\sigma_\epsilon^4} + \left(1+\frac{2G_{max,op}}{\sigma_\epsilon^2}\right)\frac{2}{\sigma_\epsilon^4}\right) + \frac{2}{\sigma_U^2}$ and $G_U := \frac{n}{2}\left(\frac{G_{max,op}}{\sigma_\epsilon^4} + \frac{1}{\sigma_\epsilon^2}\right) + \frac{1}{\sigma_U^2}$.*

Lemma 4. *$f(\boldsymbol{V},\{\boldsymbol{U}_i\}_i)$ is L_V-Lipschitz smooth with respect to $\boldsymbol{V}$ and $\|\nabla f(\boldsymbol{V},\{\boldsymbol{U}_i\}_i)\|_2 \leq G_V$ with constants $L_V := \frac{24}{\sigma_U^2}$ and $G_V := \frac{6}{\sigma_U^2}$. We use the notation $\{\boldsymbol{U}_i\}_i$ to indicate that the set is indexed over i.*

We are now ready to state our main convergence result.

Theorem 1. *Under Assumption 1, by choosing the learning rates as $\alpha = \min\{\frac{1}{2C_1G_1+L_{gu}(C_1^2G_1^2+1)}, 1\}$ and $\beta = \min\{\frac{1}{2C_2G_2+L_{gv}(C_2^2G_2^2+1)}, 1\}$, we have*

$$\min_{t=1,\dots,T}\left\{\sum_{t=1}^{T}\|g_V^t\|_F^2 + \frac{1}{m}\sum_{i=1}^{m}\|g_{U_i}^t\|_F^2\right\} \leq O\left(\frac{2\Delta_T}{\min\{\alpha,\beta\}T}\right),$$

where we define $\Delta_T = f(\boldsymbol{V}_0,\{\boldsymbol{U}_{i,0}\}_i) - f(\boldsymbol{V}_T,\{\boldsymbol{U}_{i,T}\}_i)$, $g_V^t = P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f(\boldsymbol{V}_{t-1},\{\boldsymbol{U}_{i,t}\}_i))$, $g_{U_i}^t = P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1},\boldsymbol{U}_{i,t-1}))$, C_1 and C_2 are the non-expansiveness constants of gradients, G_1 and G_2 are the bounds on the Frobenius norm of the Stiefel gradients, $L_{gu} = L_U + G_U$ and $L_{gv} = L_V + G_V$ are related to the Lipschitz constants and bounds on the gradients in Lemma 3 and 4. Theorem 1 is comparable to convergence rate of [1], which was developed for a different model (see Section I).

Remark 1. *We can obtain some insights by examining the constant terms that scale the convergence rate. Note that L_{gu} and L_{gv} are inversely dependent on σ_U^2 (see Lemma 3 and 4. Hence, more heterogeneity implies faster convergence of the algorithm in terms of training error. This can be seen in Figure 2(a). However, while less heterogeneity implies slower convergence, the convergence is to better local minimas (in terms of testing error) as seen in Figure 2(b).*

IV. Proof Outline of Theorem 1

In this section, we discuss the proof outline for Theorem 1 and preceding lemmas. First, we provide some useful facts in linear algebra and prove Lemma 3 and 4, which essentially shows that the optimization problem has nice first order properties. Then, we show the sufficient decrease properties.

Lemma 5 (Sufficient Decrease)**.** *At any iteration t, we have*

$$\begin{aligned} &f(\boldsymbol{V}_t,\{\boldsymbol{U}_{i,t}\}_i) - f(\boldsymbol{V}_{t-1},\{\boldsymbol{U}_{i,t-1}\}_i) \\ &\leq (-\alpha+C_\alpha\alpha^2)\frac{1}{m}\sum_{i=1}^{m}\|P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1},\boldsymbol{U}_{i,t-1}))\|_F^2 \\ &\quad + (-\beta+C_\beta\beta^2)\|P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f(\boldsymbol{V}_{t-1},\{\boldsymbol{U}_{i,t}\}_i))\|_F^2 \end{aligned}$$

for learning rates α, β and some cosntants C_α, C_β that are defined in the upcoming proofs.

The challenge in proving sufficient decrease on the Stiefel manifold compared to the Euclidean counterpart is that we need to ensure the projected and retracted updates are preserving the descent in the loss function, Lemma 1 is used for showing this.

A. Useful Relations and Lemmas

Before moving on with the proof outlines of the lemmas, we state some of the facts to be used in the proofs.

Fact 1. *The gradients of the local loss function with respect to the local and global PC's are given as*

$$\nabla_{\boldsymbol{U}_i} f_i(\boldsymbol{V},\boldsymbol{U}_i) = -\frac{n}{2}(\boldsymbol{W}_i^{-1}S_i\boldsymbol{W}_i^{-1}\boldsymbol{U}_i - \boldsymbol{W}_i^{-1}\boldsymbol{U}_i) + \frac{\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)}{\sigma_U^2},$$

$$\nabla_{\boldsymbol{V}} f_i(\boldsymbol{V},\boldsymbol{U}_i) = -\frac{\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)(\boldsymbol{U}_i^\top\boldsymbol{V} + \boldsymbol{V}^\top\boldsymbol{U}_i)}{2\sigma_U^2}.$$

Fact 2. *For two matrices $\boldsymbol{A} \in \mathbb{R}^{a\times b}$ and $\boldsymbol{B} \in \mathbb{R}^{b\times c}$, we have*

$$\|\boldsymbol{AB}\|_F \leq \|\boldsymbol{A}\|_{op}\|\boldsymbol{B}\|_F \textit{ and } \|\boldsymbol{AB}\|_F \leq \|\boldsymbol{A}\|_F\|\boldsymbol{B}\|_{op}.$$

Fact 3. *For matrix to matrix functions, $\{g_i\}_{i=1}^k$, with bounded output operator norms, $\max_{\boldsymbol{X}} \|g_i(\boldsymbol{X})\|_{op} \leq M_i$, we have*

$$\|\prod_{i=1}^{k} g_i(\boldsymbol{X}) - \prod_{i=1}^{k} g_i(\boldsymbol{Y})\|_F \leq \prod_{j=1}^{k} M_j \left(\sum_{i=1}^{k}\|g_i(\boldsymbol{X}) - g_i(\boldsymbol{Y})\|_F\right)$$

Proof of Lemma 3. For the bound on the gradient,

$$\begin{aligned}
&\| -\frac{n}{2}(\boldsymbol{W}_i^{-1}\boldsymbol{S}_i\boldsymbol{W}_i^{-1}\boldsymbol{U}_i - \boldsymbol{W}_i^{-1}\boldsymbol{U}_i) + \frac{\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_i)}{\sigma_U^2}\|_{op} \\
&\leq \| -\frac{n}{2}(\boldsymbol{W}_i^{-1}\boldsymbol{S}_i\boldsymbol{W}_i^{-1}\boldsymbol{U}_i - \boldsymbol{W}_i^{-1}\boldsymbol{U}_i)\|_{op} + \frac{2}{\sigma_U^2} \\
&\leq \frac{n}{2}(\|\boldsymbol{W}_i^{-1}\boldsymbol{S}_i\boldsymbol{W}_i^{-1}\boldsymbol{U}_i\|_{op} + \|\boldsymbol{W}_i^{-1}\boldsymbol{U}_i\|_{op}) + \frac{2}{\sigma_U^2} \\
&\leq \frac{n}{2}\left(\frac{G_{max,op}}{\sigma_\epsilon^4} + \frac{1}{\sigma_\epsilon^2}\right) + \frac{2}{\sigma_U^2},
\end{aligned}$$

where in the last inequality we use $\|\boldsymbol{W}_i^{-1}\|_{op} \leq \frac{1}{\sigma_\epsilon^2}$. Therefore, we find that norm of the gradient is bounded by $G_U := \frac{n}{2}(\frac{G_{max,op}}{\sigma_\epsilon^4} + \frac{1}{\sigma_\epsilon^2}) + \frac{1}{\sigma_U^2}$. For the Lipschitz continuity of the gradient, we omit the client index i and use $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ to denote two arbitrary points on $St(d,r)$ for simplicity. For any client i, we focus on the first term of the gradient,

$$\begin{aligned}
&\|\boldsymbol{W}_1^{-1}\boldsymbol{S}_i\boldsymbol{W}_1^{-1}\boldsymbol{U}_1 - \boldsymbol{W}_1^{-1}\boldsymbol{U}_1 - \boldsymbol{W}_2^{-1}\boldsymbol{S}_i\boldsymbol{W}_2^{-1}\boldsymbol{U}_2 + \boldsymbol{W}_2^{-1}\boldsymbol{U}_2\|_F \\
&\leq \Big(\frac{1}{\sigma_\epsilon^2} + \frac{G_{max,op}}{\sigma_\epsilon^4} + \Big(1 + \frac{2G_{max,op}}{\sigma_\epsilon^2}\Big)\frac{2}{\sigma_\epsilon^4}\Big)\|\boldsymbol{U}_2 - \boldsymbol{U}_1\|_F,
\end{aligned} \tag{5}$$

where we defer the algebra to [14]. For the second part of the gradient we have

$$\begin{aligned}
&\frac{1}{\sigma_U^2}\|\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_1) - \mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\boldsymbol{U}_2)\|_F \\
&= \frac{1}{\sigma_U^2}\|\boldsymbol{U}_1 - \boldsymbol{U}_2 - \frac{1}{2}\boldsymbol{V}(\boldsymbol{V}^\top(\boldsymbol{U}_1 - \boldsymbol{U}_2) + (\boldsymbol{U}_1^\top - \boldsymbol{U}_2^\top)\boldsymbol{V})\|_F \\
&\leq \frac{2}{\sigma_U^2}\|\boldsymbol{U}_1 - \boldsymbol{U}_2\|_F,
\end{aligned}$$

where in the last inequality we use Fact 2. As a result, we find that the gradient is Lipschitz continuous with $L_U := \frac{n}{2}\Big(\frac{1}{\sigma_\epsilon^2} + \frac{G_{max,op}}{\sigma_\epsilon^4} + \Big(1 + \frac{2G_{max,op}}{\sigma_\epsilon^2}\Big)\frac{2}{\sigma_\epsilon^4}\Big) + \frac{2}{\sigma_U^2}$ □

Proof of Lemma 4. In this case, it is straightforward to see that $G_V = \frac{4}{\sigma_U^2}$. For the Lipschitz constant,

$$\begin{aligned}
&\frac{2}{\sigma_U^2}\|\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}_1}}(\boldsymbol{U}_i)\mathrm{sym}(\boldsymbol{U}_i^\top\boldsymbol{V}_1) - \mathcal{P}_{\mathcal{T}_{\boldsymbol{V}_2}}(\boldsymbol{U}_i)\mathrm{sym}(\boldsymbol{U}_i^\top\boldsymbol{V}_2)\|_F \\
&= \frac{2}{\sigma_U^2}\|\boldsymbol{U}_i\boldsymbol{U}_i^\top(\boldsymbol{V}_1 - \boldsymbol{V}_2) + \boldsymbol{U}_i(\boldsymbol{V}_1 - \boldsymbol{V}_2)\boldsymbol{U}_i^\top \\
&\quad - \frac{1}{2}(\boldsymbol{V}_1(\boldsymbol{V}_1^\top\boldsymbol{U}_i + \boldsymbol{U}_i^\top\boldsymbol{V}_1) - \boldsymbol{V}_2(\boldsymbol{V}_2^\top\boldsymbol{U}_i + \boldsymbol{U}_i^\top\boldsymbol{V}_2))\|_F \\
&\leq \frac{24}{\sigma_U^2}\|\boldsymbol{V}_1 - \boldsymbol{V}_2\|_F,
\end{aligned}$$

where $\mathrm{sym}(\boldsymbol{U}_i^\top\boldsymbol{V}) = \boldsymbol{U}_i^\top\boldsymbol{V} + \boldsymbol{V}^\top\boldsymbol{U}_i$ and we used Fact 3, hence $L_V = \frac{24}{\sigma_U^2}$. □

B. Proof of Lemma 5, Sufficient Decrease

For a given iteration, we can obtain sufficient decrease in the loss function by using the previous lemmas. For the sufficient decrease with respect to $\{\boldsymbol{U}_i\}$, we have

$$\begin{aligned}
&f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t}\}_i) - f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t-1}\}_i) \\
&\leq \frac{(-\alpha + C_\alpha\alpha^2)}{m}\sum_{i=1}^m \|P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t-1}))\|_F^2
\end{aligned} \tag{6}$$

where $C_\alpha = (C_1G_1 + \frac{L_{gu}(C_1^2G_1^2+1)}{2})$. For $\boldsymbol{V}$,

$$\begin{aligned}
&f(\boldsymbol{V}_t, \{\boldsymbol{U}_{i,t}\}_i) - f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t}\}_i) \\
&\leq (-\beta + C_\beta\beta^2)\|P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t}\}_i))\|_F^2
\end{aligned} \tag{7}$$

where $C_\beta = (C_2G_2 + \frac{L_{gv}(C_2^2G_2^2+1)}{2})$. By summing (6) and (7), we obtain the overall sufficient decrease:

$$\begin{aligned}
&f(\boldsymbol{V}_t, \{\boldsymbol{U}_{i,t}\}_i) - f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t-1}\}_i) \\
&\leq (-\alpha + C_\alpha\alpha^2)\frac{1}{m}\sum_{i=1}^m \|P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t-1}))\|_F^2 \\
&\quad + (-\beta + C_\beta\beta^2)\|P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t}\}_i))\|_F^2.
\end{aligned}$$

C. Final Bound

By choosing $\alpha \leq \min\{\frac{1}{2C_\alpha}, 1\}$, $\beta \leq \min\{\frac{1}{2C_\beta}, 1\}$ and telescoping across iterations, we obtain

$$\begin{aligned}
&\frac{1}{T}\Big[\sum_{t=1}^T \|P_{\mathcal{T}_{\boldsymbol{V}_{t-1}}}(\nabla_{\boldsymbol{V}_{t-1}} f(\boldsymbol{V}_{t-1}, \{\boldsymbol{U}_{i,t}\}_i))\|_F^2 \\
&\quad + \frac{1}{m}\sum_{i=1}^m \|P_{\mathcal{T}_{\boldsymbol{U}_{i,t-1}}}(\nabla_{\boldsymbol{U}_{i,t-1}} f_i(\boldsymbol{V}_{t-1}, \boldsymbol{U}_{i,t-1}))\|_F^2\Big] \\
&\leq \frac{2(f(\boldsymbol{V}_0, \{\boldsymbol{U}_{i,0}\}_i) - f(\boldsymbol{V}_T, \{\boldsymbol{U}_{i,T}\}_i))}{T\min\{\alpha, \beta\}}.
\end{aligned}$$

Taking the minimum of all iterations directly gives Theorem 1.

V. Experiments

In this section, we compare the reconstruction error of *our algorithm*, *local training*, and *global averaging* through synthetic datasets, where in global averaging, the gradients were gathered before performing projection and retraction. We will see that our algorithm acts like an interpolation between the global averaging and local training, and performs the best in terms of reconstruction error in different scenarios.

First, we show the generation of our synthetic datasets. For any given $\boldsymbol{V} \in \mathbb{R}^{d\times r}$, we generate $\{\widetilde{\boldsymbol{U}}_i\}_{i=1}^m$ such that each entries of $\widetilde{\boldsymbol{U}}_i \in \mathbb{R}^{d\times r}$ is sampled as i.i.d. Gaussian, $(\widetilde{\boldsymbol{U}}_i)_{jk} \overset{i.i.d.}{\sim} \mathcal{N}(\boldsymbol{V}_{jk}, \sigma_U^2)$ for all $i \in [m] \overset{\text{def}}{=} \{1, 2, \ldots, m\}$. Then, the $\widetilde{\boldsymbol{U}}_i$'s are projected onto the tangent space of $\boldsymbol{V}$ and retracted onto the Stiefel manifold through polar retraction to generate the underlying PC matrices, $\boldsymbol{U}_i$'s, that is $\boldsymbol{U}_i = \mathcal{R}_{\boldsymbol{V}}(\mathcal{P}_{\mathcal{T}_{\boldsymbol{V}}}(\widetilde{\boldsymbol{U}}_i))$ for all $i \in [m]$. Then, we generate the data $\boldsymbol{Y}_i \in \mathbb{R}^{d\times n}$ as we stated in Section II.

We apply our algorithm, global averaging, and local training on synthetic datasets and compare their reconstruction error defined as $\frac{1}{mn}\sum_{i=1}^m \|\boldsymbol{Y}_i - \boldsymbol{U}_{i,t}\boldsymbol{U}_{i,t}^\top\boldsymbol{Y}_i\|_F^2$. In the following numerical results, we will show the ratio of the reconstruction error to the true reconstruction error, where the true reconstruction error is defined as $\frac{1}{m}\sum_{i=1}^m \mathbb{E}_{\boldsymbol{y}_i}\Big[\|\boldsymbol{y}_i - \boldsymbol{U}_i\boldsymbol{U}_i^\top\boldsymbol{y}_i\|_F^2\Big]$.

Convergence. First, we show the convergence result of our algorithms, global averaging, and local training. In Figure 1(a), local training converges the fastest while attaining the worst performance. On the other hand, our algorithm performs the best while yielding a longer convergence time. In general, global averaging does not always outperform local training.

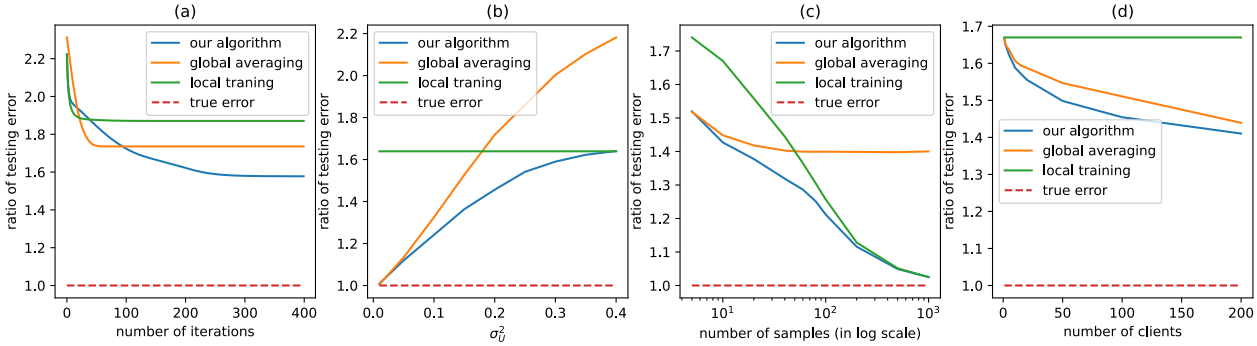

Fig. 1. The y-axis shows the ratio of the reconstruction error to the true reconstruction error defined earlier in this section. (a) Ratio of error to the number of iterations. The parameters are $(d, r) = (30, 10)$, $m = 200$, $n = 10$, $\sigma_U^2 = 0.2$, $\sigma_X^2 = 1$, and $\sigma_\epsilon^2 = 0.5$. The initial PCs are generated uniformly on the Stiefel manifold and the reconstruction error is averaged over 10 different datasets. (b) Ratio of error to σ_U^2. The parameters are $(d, r) = (30, 10)$, $m = 200$, $n = 20$, $\sigma_X^2 = 1$, and $\sigma_\epsilon^2 = 0.5$. The reconstruction error is averaged over 10 different datasets. (c) Ratio of error to the number of samples, n. The parameters are $(d, r) = (100, 20)$, $m = 100$, $\sigma_U^2 = 0.1$, $\sigma_X^2 = 1$, and $\sigma_\epsilon^2 = 0.5$. The reconstruction error is averaged over 20 different datasets. (d) Ratio of error to the number of clients, m. The parameters are $(d, r) = (100, 20)$, $n = 10$, $\sigma_U^2 = 0.1$, $\sigma_X^2 = 1$, and $\sigma_\epsilon^2 = 0.5$.

As we will see in the following results, global averaging and local training have their own preferable regimes whereas our algorithm always attains the smallest reconstruction error.

Dependence on σ_U^2. Figure 1(b) shows the relation between the reconstruction error and the heterogeneity factor, σ_U^2. When σ_U^2 is small, the $\boldsymbol{U}_i$'s are close to $\boldsymbol{V}$. The datasets are highly homogeneous and thus both our algorithm and global averaging hugely benefit from the collaboration between the clients and attain low reconstruction error. When σ_U^2 is large, the level of heterogeneity between the datasets is high and naively applying global averaging leads to a poor result. Meanwhile, in our algorithm, the effect of regularization becomes smaller as σ_U^2 increases. Thus, our algorithm performs like local training and attains a similar error to pure local training. This shows that our algorithm acts like an interpolation of global averaging and pure local training. Our algorithm performs similarly to one of them in extreme cases and outperforms them for a medium level of heterogeneity.

Dependence on the number of local samples. Figure 1(c) shows the relation between the reconstruction error and the number of samples on each client, n. For global averaging, the reconstruction error does not converge to the true error as the number of samples increases since more samples overcome the effect of noise but not heterogeneity. On the other hand, the local training is not affected by heterogeneity. The error decreases as n increases and eventually outperforms global averaging. Our algorithm plays as an interpolation between the two. When each client has a small amount of samples, the regularization term enforces collaboration between the clients to attain a smaller reconstruction error. When each client has a sufficient amount of samples, we can see from the gradients that the regularization has a much smaller impact and our algorithm acts like local training.

Dependence on the number of clients. Figure 1(d) shows the relation between the reconstruction error and the number of clients, m. The reconstruction error for local training remains the same, while for both our algorithm and global averaging, the error decreases as the number of clients increases.

Convergence rate as a function of σ_U^2. In Remark 1, we mentioned that more heterogeneity implies faster convergence

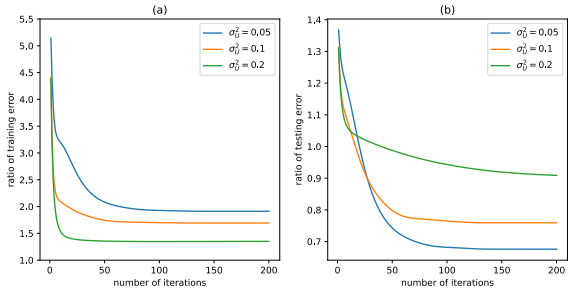

Fig. 2. Comparing the convergence rate of training error for different σ_U^2's. The parameters are $(d, r) = (30, 10)$, $m = 50$, $n = 20$, $\sigma_U^2 = 0.2$, $\sigma_X^2 = 1$, and $\sigma_\epsilon^2 = 0.5$. The learning rate are tuned to be $(0.02, 0.04, 0.04)$. The error is averaged over 20 different datasets.

of the algorithm in terms of training error. In Figure 2, we show the convergence performance for different σ_U^2's. Note that we plot training and testing error in Figure 2(a) and Figure 2(b), respectively. For each σ_U^2, we tuned the learning rate so that larger learning rates lead to worse training error and smaller learning rates lead to slower convergence in training error. We can see that a larger σ_U^2 indeed leads to a faster training convergence in Figure 2(a). It attains a smaller training error since our algorithm is essentially doing local training for larger σ_U^2 and thus is more likely to overfit. However, Figure 2(b) shows that larger σ_U^2 still leads to a larger testing error despite its training performance.

VI. Conclusion

In this work, we proposed a hierarchical Bayes statistical formulation to model personalized PCA algorithms. Our formulation lead to an optimization problem that we proposed an alternating Stiefel gradient descent algorithm to solve. We analyzed the convergence properties of the resulting algorithm and deduced a relationship between convergence and heterogeneity of the federated ecosystem. Finally, we have shown the effectiveness of our proposed algorithm by comparing to competing methods on synthetic datasets.

For future work, we are planning to explore different prior distributions that is suitable for diverse applications. Moreover, we are planning to extend our experiments to real world data and explore the role multiple local iterations in the algorithm.

REFERENCES

[1] N. Shi and R. A. Kontar, "Personalized pca: Decoupling shared and unique features," *arXiv preprint arXiv:2207.08041*, 2022.

[2] A. Grammenos, R. Mendoza Smith, J. Crowcroft, and C. Mascolo, "Federated principal component analysis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6453–6464. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/47a658229eb2368a99f1d032c8848542-Paper.pdf

[3] D. Garber, O. Shamir, and N. Srebro, "Communication-efficient algorithms for distributed stochastic principal component analysis," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1203–1212. [Online]. Available: https://proceedings.mlr.press/v70/garber17a.html

[4] L.-K. Huang and S. Pan, "Communication-efficient distributed PCA by Riemannian optimization," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 4465–4474. [Online]. Available: https://proceedings.mlr.press/v119/huang20e.html

[5] F. Alimisis, P. Davies, B. Vandereycken, and D. Alistarh, "Distributed principal component analysis with limited communication," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=edCFRvlWqV

[6] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," in *Advances in Neural Information Processing Systems*, 2020.

[7] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," in *Advances in Neural Information Processing Systems*, 2020.

[8] 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.

[9] K. Ozkara, N. Singh, D. Data, and S. Diggavi, "Quped: Quantized personalization via distillation with applications to federated learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[10] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," in *International Conference on Learning Representations*, 2021.

[11] K. Ozkara, A. Girgis, D. Data, and S. Diggavi, "A statistical framework for personalized federated learning and estimation: Theory, algorithms, and privacy," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=FUiDMCr_W4o

[12] K. Ozkara, A. M. Girgis, D. Data, and S. Diggavi, "A generative framework for personalized learning and estimation: Theory, algorithms, and privacy," *arXiv preprint arXiv:2207.01771*, 2022.

[13] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[14] K. Ozkara, B. Huang, and S. Diggavi, "Personalized pca for federated heterogeneous data," *Available on arXiv*, 2023.

[15] T. Kaneko, S. Fiori, and T. Tanaka, "Empirical arithmetic averaging over the compact stiefel manifold," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 883–894, 2012.

[16] S. Chen, A. Garcia, M. Hong, and S. Shahrampour, "Decentralized riemannian gradient descent on the stiefel manifold," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1594–1605.