Creating Coherence in Federated Non-Negative Matrix Factorization

Sebastian Dalleiger, Aristides Gionis

KTH Royal Institute of Technology, Stockhom, Sweden
sdall@kth.se, argioni@kth.se

Abstract

In many real-world applications, data is inherently decentralized, necessitating data analysis methods that prioritize privacy while delivering interpretable results. Federated Non-Negative Matrix Factorization (FedNMF) meets this requirement by factorizing latent components from distributed data that cannot be freely shared among clients. A significant challenge in FedNMF arises when clients converge on different solutions due to prolonged independent optimization, leading to drift and incoherent models. While Federated Learning (FL) typically mitigates drift through frequent synchronizations and strong regularization, it often overlooks critical properties of Non-Negative Matrix Factorization, such as permutation invariance. As a result, solutions from FedNMF clients may be misidentified by FL drift as distinct, despite being equivalent. Using an alignment-aware drift, we create coherence through proximal optimization and barycenter aggregation for FedNMF. We analyze the computational complexity of our approach, provide efficient heuristics, and ensure the convergence of our algorithms. On a diverse set of real-world and synthetic datasets, we demonstrate the effectiveness of our methods.

Replication Material — doi.org/10.5281/zenodo.14501532

1 Introduction

Uncovering hidden structure and relationships within distributed non-negative data is a prevalent challenge in various domains, including life sciences (Xu et al. 2021), collaborative filtering systems (Yang et al. 2021), and topic modeling (Wang, Tong, and Shi 2020). In many instances, this non-negative data is distributed across different devices, and sharing raw data is restricted due to privacy concerns. For instance, in life science, individual hospitals may possess non-negative matrices representing patient-related information, and privacy regulations mandate that these matrices cannot be shared directly.

Federated learning (FL) is a decentralized machine-learning paradigm that enables multiple clients to collaboratively train a global model while keeping their data local and private. By allowing for efficient distributed optimization, federated learning thus enables learning from sensitive data and enhances privacy. Those are sought-after properties in areas that deal with large quantities of sensitive data, such as healthcare, where Non-Negative Matrix Factorization (NMF) is a prevalent algorithm. Extending NMF, Federated NMF (FedNMF) aims to factorize a distributed matrix $X_j \approx U_j \bar{V}$ into a local set of basis matrices $U_j \in \mathbb{R}_+^{n_j \times k}$ and a globally-shared coefficient matrix $\bar{V} \in \mathbb{R}_+^{k \times m}$. To efficiently solve this problem in parallel, we independently optimize local auxiliary problems of the form $X_j \approx U_j V_j$ and periodically aggregate the results into a shared global matrix

$$\bar{V} \leftarrow \textsc{Aggregate}(V_1, \ldots, V_c) \ .$$

This approach is effective when the objective remains relatively stable once we replace V_j with $\bar{V}$, i.e., $U_j \bar{V} \approx U_j V_j$.

Whenever local matrices converge to the same solution, aggregating such a coherent model is easy. However, if local matrices diverge to different solutions, aggregating a global coherent model is hard. Usually, as clients continue to update their local matrices, their matrices become increasingly misaligned with the global state. We call this phenomenon coefficient drift, which is often defined as the sum

$$\sum_j^c \|V_j^{t+\tau} - \bar{V}^{t+\tau}\|_{\mathrm{F}}^2$$

of distances between the global $\bar{V}$ and local V_j after τ client optimization steps. A high drift imposes significant challenges during learning. First, a high drift significantly deteriorates the quality of the average model because it increases variance. Second, a high drift drift slows convergence both locally and globally, because synchronizations may lead to substantial changes in the model's coefficients. Third, a high drift can result in the global model not fitting the local-data distribution, as some clients may be overshadowed by the majority. Finally, drift may be exacerbated when local models are optimized independently for extended periods. Therefore, preventing drift and thereby creating coherence is a core aspect of efficient FL.

Generic techniques such as frequent model-synchronization (McMahan et al. 2017), strong regularization (Li et al. 2020; Yuan and Li 2022), and gradient control (Karimireddy et al. 2020) might help with maintaining model alignment and improve the overall client-server coherence. Although these approaches are successful in the realm of federated machine learning, standard federated learning

techniques often overlook the semantics of NMF. At its core, the *definition* of drift introduces a hidden assumption which directly contradict core aspects of matrix-factorization algorithms. While the usual drift assumes that local and global models are *naturally aligned*, matrix factorization clients seek models that are *intrinsically permutation invariant*. Consequently, generic federated learning techniques may aggregate incorrect coefficients and impose proximity constraints between misaligned models, leading to significant performance degradation.

In matrix factorization, rows in $\bar{V}$ and V_j represent *components* which capture patterns within the data. Matrix factorization clients are allowed to encode identical or similar components in different locations then other clients. As a result, even if local matrices are equivalent under permutation, the usual federated learning drift is still substantial. To address this challenge, generic federated learning must implement frequent periodic model synchronizations, which introduces undesirable and unnecessary communication overhead when applied to FedNMF. Furthermore, NMF clients may identify components that are relevant in a local context but lack global significance. However, the usual drift assumes a one-to-one correspondence between local and global components. Therefore, by preventing the typical drift we also enforce clients to overwrite locally-relevant components, thereby inhibiting local personalization.

To alleviate these problems without unnecessary communication overhead:

1) We introduce an adaptable model-alignment framework for federated learning that allows for easy switching of alignment strategies and personalized federated learning.
2) We create coherent clients by projecting local models towards the global model using a proximal operator for alignment-aware drift.
3) We formulate an alternating local optimization within federated learning, establishing a necessary convergence property for successful algorithms.
4) We introduce matrix barycenters for server aggregation, analyze their computational complexity, and provide efficient algorithms, enhancing standard averaging practices.
5) We experimentally demonstrate the effectiveness and limitations of our approach on a wide-range of real-world and synthetic data.

2 Related Work

Our work integrates three research domains to tackle the challenges of federated matrix factorization.

In the context of **matrix factorization**, our approach is a decentralized adaptation of non-negative matrix factorization (NMF) (Paatero and Tapper 1994; Lee and Seung 1999, 2000). Having partitioned matrices, our method is related to multi-view joint NMF (MV-NMF) (Liu et al. 2013) and collective NMF (CNMF) (Singh and Gordon 2008). Notably, unlike these algorithms, ours is federated and capable of operating under strict data-access restrictions.

Our alignment framework lies in the realm of **assignment problems** and barycenters. We capture various assignment problems, such as complete and incomplete linear assignment problems (Kuhn 1955; Edmonds and Karp 1972) and stochastic assignments (Cuturi 2013), thus spanning multiple federated scenarios with one framework. Our barycenter is inspired by Wasserstein barycenters (Luise et al. 2019) for probability distributions. However, rather than only transporting probability mass, our barycenter operates with various different alignment algorithms, whereby we cover barycenters for exact, incomplete, and stochastic alignments.

In the domain of **federated learning**, our approach adheres to the principle of learning models by leveraging centralized aggregations of decentralized model coefficients (McMahan et al. 2017). Similar to FedProx (Li et al. 2020) and MV-NMF, we employ a proximal penalty to maintain coherence. However, we specialize in NMF and exploit permutation invariance directly to maintain coherence. Recent work has explored the intersection between federated learning and optimal transport, e.g., in personalized federated learning (Farnia et al. 2022), but it does not study permuting coefficient components and does not allow for exact or incomplete alignments, unlike our work. Federated NMF has found applications in federated topic modeling (Si et al. 2022) and federated clustering (Wang and Chang 2022), utilizing traditional federated learning approaches without addressing alignment strategies under unrestricted access to basis matrices. The recent advancements with a focus on Differential Privacy in Federated Learning and Non-negative Matrix Factorization (Li et al. 2021) illustrate the potential for addressing federated factorization without compromising data privacy, but these algorithms focus on different aspects of FL without specifically targeting coherence through alignment-awareness.

3 Creating Coherence

Having introduced our problem, we now describe our solution strategies, starting with an introduction to Federated Non-Negative Matrix Factorization (FedNMF). We are given data that is *distributed* across c clients

$$X = \left[X_1, X_2, \cdots X_c\right]^\top \ ,$$

where each local matrix X_i is *secret* and cannot be shared with other clients and servers. Each client j aims to factorize X_i as the product of a secret *local basis* matrix $U_i \in \mathbb{R}^{n_i \times k}$ and a *global coefficient* matrix $\bar{V} \in \mathbb{R}^{k \times m}$. Clients are prohibited from sharing X_i and U_i due of privacy restrictions. All clients are, however, granted access to $\bar{V}$, whose minimizer they collaboratively seek to identify.

A common approach to NMF optimizes factor matrices through block-coordinate descent, aiming to minimize the reconstruction loss. This involves alternately taking gradient steps while keeping the others fixed. In FedNMF, all clients are expected to share a common matrix $\bar{V}$. Since using such an alternating optimization on $\bar{V}$ would necessitate a serialized update of $\bar{V}$, processing one client at a time, we would not only prevent parallelization, but also introduce a significant communication overhead among clients. To alleviate this, we turn to federated learning, where we independently factorize subproblems in parallel and aggregate the outcome into a global solution. In particular, we solve an independent subproblem $X_i \approx U_i V_i$ at each client using alternating opti-

mization. Then, we aggregate all V_i's into a global matrix $\bar{V}$ at the server.

This aggregation of $V_1, \ldots, V_c$ into a global $\bar{V}$ works best when V_i's have low variances and are similar to another. Informally, we call a sufficiently small distance between $\bar{V}$ and all $V_i \in \mathcal{V}$ coherence. Because clients minimize their own losses independently, they capture local particularities, patterns, and characteristics from their data distributions which differ from the rest. When clients optimize for a prolonged time, they create matrices V_i that diverge from other client matrices and from $\bar{V}$, thereby causing incoherence. In other words, incoherence stems from local matrices that drift from global matrices during optimization.

Definition 1 (Regular Coefficient Drift)**.** For any discrete time $t, \tau \in \mathbb{N}$, for each client $i \in \mathbb{N}$, drift is the distance

$$\|V_i^{t+\tau} - \bar{V}^t\|_{\mathrm{F}}^2 \,. \tag{1}$$

between global matrix $\bar{V}^t$ and local matrix $V_i^{t+\tau}$. Moreover, if sufficiently small, then we have coherence.

As incoherence is a major obstacle to efficacy in FL, similar definitions have found wide applicability (Li et al. 2020) to maintain coherent machine-learning models. However, this notion introduces unwanted assumptions that are fundamentally incompatible with matrix factorization. In NMF, features possess a joint intrinsic meaning and collectively represent patterns within the data. These *components* correspond to the rows in the matrices $V_1, \ldots, V_c$ and $\bar{V}$. Traditional FL drift and coherence notions assume that clients encode components with identical semantics in the same position $l \in [k]$ where $[V_i]_l \approx [V_j]_l$ and $[V_i]_l \approx \bar{V}_l$ for all clients $i, j \in [c]$. However, matrix factorization is inherently permutable, which means that components can be encoded in different positions like $[V_i]_l \approx [V_j]_{l'}$ $l' \neq l$. Consequently, the standard drift assumption is not meaningfully applicable to FedNMF without sacrificing its semantics.

We address this issue by introducing a new *alignment-aware drift*. At the core of this notion lies a mechanism that aligns components with their best-matching counterparts. Rather than considering the regular distance between matrices, we consider a matching distance which first *optimally aligns* components, and then computes the distance between aligned matrices.

Definition 2 (Matching Distance Framework)**.** We define the minimum matching distance between X and Y as

$$D_p(X, Y) \triangleq \min_P \tfrac{1}{2}\|X - PY\|_{\mathrm{F}}^2 \tag{2}$$

subject to different sets of constraints below.

To address the requirements of three key scenarios in federated matrix factorization, we specialize $D_p(V_j, \bar{V})$ by applying different sets of constraints to alignment matrix P.

Without Personalization In the most common scenario, we aim to align components in the global matrix $\bar{V}$ with the local matrices V_j exactly. To adapt Def. 2 to this context, we ensure that each component in $\bar{V}$ is precisely aligned with one in V_j by requiring that P is a permutation matrix with

$$D_1 : P \in \{0,1\}^{k\times k}, \quad P^\top P = \mathbb{I} \,.$$

We compute these matrices P by solving its corresponding *Linear Assignment Problem* (LAP), using the Hungarian algorithm in $O(k^3)$ time (Kuhn 1955; Edmonds and Karp 1972). While common, this exactness prevents clients from estimating local data particularities, which we alleviate next.

With Personalization As before, we align components in $\bar{V}$ with those in V_j. However, we now exclude local components in any V_j from the alignment that are deemed globally irrelevant. A component is considered globally irrelevant if it is sufficiently distant from all components in $\bar{V}$. To determine irrelevancy, we employ the *correlation distance* $\mathrm{d}^\rho = 1 - \mathrm{cor}(X, Y)$ detailed in Apx. B. To make a statistically sound decision regarding what means to be "sufficiently far away", we stretch distances d_{ij}^ρ to infinity whenever i and j are *insignificantly* correlated, assessed by an z-test, yielding

$$D_2 \colon P \in \{0,1\}^{k\times k}, P\,\mathbf{1}_k = \{0,1\}_k, P_{ij} = 1 \iff d_{ij}^\rho < \infty \,.$$

This specialization also remains efficiently computable using an auxiliary LAP problem that includes an *unaligned* node with the largest yet finite cost to reach.

Relaxed Personalization Finally, instead of enforcing a strict one-to-one correspondence between components, we adapt our alignment strategy to allow for *one-to-many* stochastic alignments. For this, we align each component in any local matrix V_j to a convex combination of components from the global matrix $\bar{V}$, formalized as

$$D_3 \colon P \in [0,1]^{k\times k}, \quad P\,\mathbf{1}_k = \mathbf{1}_k \,.$$

Here, P represents the weights of the stochastic alignments, resulting in a transport problem for which we employ a *Sinkhorn algorithm* (Cuturi 2013).

Client

Having described our matching-distance framework, we now introduce this concept to our clients. Ideally, we would like to always have zero drift $D_p(V_j, \bar{V}) = 0$, but this would require an inefficient round-robin-style serial optimization. To empower efficient distributed optimization and local personalization, we decouple clients and optimize

$$\Phi(U, V) \triangleq \textstyle\sum_j \Phi_j(U_i, V_i) \,,$$

in parallel, based on private data X_j, secret basis matrix U_j, and sharable coefficient matrix V_j. Like in regular NMF, our clients seek to minimize the distance between reconstruction $U_j V_j$ and data X_j. To create coherence, we additionally regularize against drift with D_p, yielding the local objective

$$\textsc{client}\ \ \Phi_j(U_j, V_j) \triangleq \tfrac{1}{2}\|X_j - U_j V_j\|_{\mathrm{F}}^2 + \gamma D_p(V_j, \bar{V}) \,,$$

for non-negative $U_j \geq 0$, $V_j \geq 0$. Following the state-of-the-art, each client optimizes using an alternating minimization scheme, specifically *inertial proximal alternating linear minimization* (iPALM) (Bolte, Sabach, and Teboulle 2014), where we alternate between the update rules

$$U_j \leftarrow \left[U_j - \eta_{U_j} \nabla_{U_j} \tfrac{1}{2}\|X_j - U_j V_j\|_{\mathrm{F}}^2\right]_+ \text{ and}$$
$$V_j \leftarrow \widehat{\mathrm{prox}}_{\eta_{V_j}}^{D_p(\cdot,\bar{V})}\left[V_j - \eta_{V_j} \nabla_{V_j} \tfrac{1}{2}\|X_j - U_j V_j\|_{\mathrm{F}}^2\right]_+ \,,$$

Algorithm 1: Fixed-point Barycenter

Input: matrices $\mathcal{V}$ (with $V \in \mathbb{R}^{k\times m}\ \forall V \in \mathcal{V}$), number of iterations $T \in \mathbb{N}$
Output: barycenter $\bar{V}$, alignment-plans $\mathcal{P}$
$\bar{V} \leftarrow \frac{1}{|\mathcal{V}|}\sum_i V_i$
$\mathcal{P} \leftarrow \{\mathbb{I}_k \mid \forall V \in \mathcal{V}\}$
for $t = 1, \ldots, T$ **or until convergence do**
  $P_j \leftarrow \arg D_p(\bar{V}, V_j) \forall j$
  $\bar{V} \leftarrow \frac{1}{|\mathcal{P}|}\sum_j P_j V_j$
end

until convergence or for a fixed number of steps, utilizing the $\widehat{\text{prox}}^{D_p}$ operator to maintain proximity to $\bar{V}$ in Eq. (3) below, together with the $[X]_+ \triangleq \max\{X, 0\}$ proximal operator to ensure non-negativity.

Maintaining coherence Although inspired by FEDPROX (Li et al. 2020), unlike FEDPROX, we do not use a regularizer during optimization to maintain proximity to $\bar{V}$. Instead, we identify the closest matrix to V_j in alignment-aware proximity to the matrix $\bar{V}$, by projecting gradients using our *proximal operator*

$$\text{prox}_t^{D_p}(V_j) \triangleq \arg\min_A {}^1\!/_t \|A - V_j\|_{\text{F}} + \gamma D_p(A, \bar{V}) \ . \quad (3)$$

This operator involves a joint minimization problem over A and P, which does not lend itself for efficient optimization. Noting that its argument A and V_j are by definition in proximity, we compute the operator $\hat{P} \leftarrow \arg D_p(V_j, \bar{V})$ for V_j instead, thus simplifying the operator into a weighted average

$$\widehat{\text{prox}}_t^{D_p}(V_j) = [1+\gamma]^{-1}[\gamma \hat{P}\bar{V} + V_j]$$

of V_j and the V_j-aligned $\hat{P}\bar{V}$ matrix.

Finally, we obtain our client algorithm Alg. 2 by combining the above with our gradients

$$\nabla_{U_j} = U_j V_j V_j^\top - X_j V_j^\top \text{ and } \nabla_{V_j} = U_j^\top U_j V_j - U_j^\top X_j \ .$$

For the optimization rate, we use the Lipschitz constants of our gradients $\eta_{V_j}^{-1} = \|U_j U_j^\top\|$ and $\eta_{U_j}^{-1} = \|V_j^\top V_j\|$ (Bolte, Sabach, and Teboulle 2014).

Server

Having described how clients create coherence with the server, we now turn to the server's role in aggregating matrices. The server's goal is to aggregate local matrices into a global matrix, which is usually done by averaging local matrices $V_1, \ldots, V_c$. However, averaging is not suitable in the context of matrix factorization, where client matrices must be aligned. To address this, we aim to jointly identify the optimal global matrix and optimally aligned local matrices. In other words, we seek to determine the "center of mass" or barycenter.

Definition 3 (Barycenter)**.** The *barycenter* of a finite set of matrices $\mathcal{V} = \{V_j \in \mathbb{R}^{n\times m}, \ldots\}$ is the minimizer

$$\text{B}_p(\mathcal{V}) \triangleq \arg\min_{\bar{V}\in\mathbb{R}^{n\times m}} \sum_j D_p(\bar{V}, V_j) \ . \quad (4)$$

Algorithm 2: FNMF: Federated NMF

Input: matrices $\mathcal{X} = \{X_j\}_{j\in[c]}$ with $X_j \in \mathbb{R}_+^{n_j\times m}$
Input: targeted number of components $k \in \mathbb{N}$
Output: globally-shared coefficient matrix $\bar{V} \in \mathbb{R}_+^{k\times m}$
$\mathcal{U} \leftarrow \{U_j \sim \mathbb{U}_{n_j\times k} \mid \forall X_j \in \mathcal{X}\}$
$\mathcal{V} \leftarrow \{V_j \sim \mathbb{U}_{k\times m} \mid \forall X_j \in \mathcal{X}\}$
$\bar{V} \leftarrow \frac{1}{|\mathcal{X}|}\sum V_j$
repeat
  for $j = 1, \ldots, |\mathcal{X}|$ **in parallel do**
    CLIENT$(X_j, U_j, V_j, \bar{V})$
  end
  $\bar{V}, \mathcal{V} \leftarrow$ SERVER$(\mathcal{V})$
until convergence
function CLIENT$(X, U, V, \bar{V})$
  for $t = 1, \ldots, T$ **or until convergence do**
    $U \leftarrow \left[U - \eta_U \nabla_U \frac{1}{2}\|X - UV\|_{\text{F}}^2\right]_+$
    $V \leftarrow \left[V - \eta_V \nabla_V \frac{1}{2}\|X - UV\|_{\text{F}}^2\right]_+$
    $V \leftarrow \widehat{\text{prox}}_{\eta_V}^{D_p(\cdot,\bar{V})}(V)$
  end
  if differentially private **then**
    $V \leftarrow V \oplus N,\ N \sim \mathcal{N}(0, \sigma)_{k\times m}$
  end
end
function SERVER$(\mathcal{V})$
  $\bar{V} \leftarrow$ BARYCENTER$(\mathcal{V})$
  $V_j \leftarrow \bar{V}\ \forall\, V_j \in \mathcal{V}$ *(synchronize clients)*
  return $\bar{V}, \mathcal{V}$
end

Theorem 1 (Computational Complexity)**.** *The barycenter problem in Def. 3 is NP-hard. Thus, no polynomial-time algorithm is known for it, assuming $P \neq NP$.*

Proof. To show NP-hardness, we start with reducing the unconstrained barycenter probem $\arg\min_{\bar{V}\in\mathbb{R}^{n\times m}} \sum_{j=1}^n \min_{P_j} \|\bar{V} - P_j V_j\|_F^2$. to k-means (Garey and Johnson 1979). By simply stacking matrices V_j $v_i = \left[V_1, V_2, \ldots . V_c\right]_i^\top$. we obtain a k-means objective $\arg\min_{\bar{V},W} \sum_{i=1}^n \sum_{j=1}^k W_{ij}\|v_i - \bar{V}_j\|_F^2$. where the assignment matrix $W_{ij} = \mathbf{1}\{j = \arg\min_{j'} \|v_i - \bar{V}_{j'}\|\}$ alignes centroids $\bar{V}_j$ with rows v_i. Note that neither adding permutation constraints for both D_1 and D_2, nor incorporating stochasticity constraints for D_3 will reduce the complexity class. □

Analogous to the Wasserstein barycenter (Altschuler and Boix-Adserà 2022), the matrix barycenter problem is NP-hard, making exact solutions difficult to obtain. Therefore, we resort to heuristics. The first approach that comes to mind is employing k-means heuristics, such as Lloyd's algorithm (1982). While fast in practice, k-means disregards our constraints and often yields empirically lower-quality solutions as demonstrated in Sec. 4. Therefore, we introduce a fixed-point algorithm in Alg. 1, a heuristic algorithm in Apx. A (Alg. 3), and a hierarchical decomposition method in Apx. A (Alg. 4). In a nutshell, our fixed-point iteration repeatedly aligns V_j with the current barycenter and then averages the

aligned matrices to produce a new barycenter, continuing this process until convergence. The server receives each V_i from all clients and computes the barycenter as

$$\text{SERVER} \quad \bar{V} \leftarrow \mathrm{B}_p(V_1, \ldots, V_c) \ .$$

Federated NMF

Combining clients and server, we present FNMF, our algorithm for federated non-negative matrix factorization, as Alg. 2. In summary, each client factorizes its data using an alternating proximal-gradient optimization procedure. Once all clients have completed their local computations, the server aggregates the results by computing $\bar{V} \leftarrow \mathrm{B}_p(V_1, \ldots, V_c)$ and synchronizes this aggregate with each client, updating $V_j \leftarrow \bar{V}$. This process is repeated until the global objective converges, as detailed below.

Theorem 2 (Convergence)**.** *The objective $\Phi(z^t)$ converges to a stable solution and with value Φ^* of Φ as $t \to \infty$ for the sequence $\{z^t \triangleq (\{U_i^t\}_i, \{V_i^t\}_i, \bar{V}^t)\}_{t \in \mathbb{N}}$ generated by Alg. 2.*

Proof. (Sketch, full proof in Apx. C). We show the convergence of the objective to a stable solution Φ^* by first ensuring the convergence of each client by identifying *sufficient reduction* in local objectives and a *bounded dissimilarity* to $\bar{V}$. Building on this, we establish global convergence by proving that the global loss gradient is bounded by a *diminishing term* under barycenter aggregation at the server, showing that $\Phi(z^t)$ converges to a constant Φ^* as t approaches infinity. □

Guaranteeing Differential Privacy With established convergence, we now guarantee differential privacy. Our approaches ensures that no direct relationships between data points and features are disclosed to the server. However, an adversary may still try to derive sensitive information from the coefficients V_i. We prevent this by ensuring differential privacy, where each client adds noise to V_j before transmission. That is, we achieve (ϵ, δ)-differential privacy using a Gaussian noise mechanism as follows.

Definition 4 (Dwork, Roth et al. (2014))**.** A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is $(\epsilon, \delta > 0)$-differentially private (DP) if $\mathbb{P}(\mathcal{A}(X) \in S) \leq e^{\epsilon}\, \mathbb{P}(\mathcal{A}(X') \in S) + \delta$ holds for each subset $S \subset \mathcal{Y}$ and all pairs X, X' of data neighbors.

To ensure (ϵ, δ)-DP for $\sigma = \Delta\epsilon^{-1}\sqrt{2\log(5/(4\delta))}$ (Balle and Wang 2018), we apply Gaussian noise with mean 0 and variance σ to the local coefficients V_i before transmission, where $\Delta = \sup_{X,X'} \|\mathcal{A}(X) - \mathcal{A}(X')\|$ is the sensitivity of the algorithm $\mathcal{A}$. Similarly, adding Laplacian noise with mean 0 and variance $\Delta\epsilon^{-1}$ achieves $(\epsilon, 0)$-differential privacy (Dwork et al. 2006).

4 Experiments

Having introduced our algorithms, we now systematically evaluate their practical performance. We implemented our methods in the Julia programming language and ran experiments on 16 cores of an AMD EPYC 7702 CPU and a single NVIDIA A40 GPU, reporting wall-clock time. We provide the source code, datasets, synthetic dataset generator, hyperparameters, and other information needed for reproducibility.[1]

[1] https://doi.org/10.5281/zenodo.14501532

Federated Learning for NMF FL often relies on stochastic gradient descent (SGD). However, NMF typically exhibits superior performance when using deterministic, specialized alternating optimization procedures. Consequently, applying standard FL methods to NMF may yield performance characteristics that are not reminiscent for the ideas behind different FL approaches. To facilitate a fair comparison between these concepts, it is essential to reconcile FL and NMF. We achieve this by integrating state-of-the-art algorithms for both NMF and FL where applicable. Primarily, we utilize the inertial proximal alternating linear minimization (iPALM) and Multiplicative Update Rules (MUR) for NMF, adapting these algorithms to a federated context by incorporating key ideas from FedAvg and FedProx for FL. This leads to a *set of novel specialized Federated NMF algorithms*. In summary, we compare the following approaches.

- FNMF (Alg. 2), which employs different context-dependent alignment algorithms.
- FedAvg (McMahan et al. 2017), which utilizes iPALM with FedAvg's averaging.
- FedProx (Li et al. 2020), which combines iPALM under FedProx's regularization and averaging.
- iPALM (Pock and Sabach 2016), which averages independent iPALM-based NMF results once.
- MUR (Lee and Seung 1999), which averages independent MUR-based NMF results once.

In the following experiments, we ask five questions:

1) How precisely do we estimate barycenters?
2) How swiftly does our methods converge under drift?
3) How robust do our clients reconstruct target matrices?
4) How well does our method scale?
5) How resilient is our method under differential privacy?

Additional questions are addressed in Apx. D. To answer our questions, we generate synthetic data with known ground truth by filling matrices of various sizes with randomly-placed dense blocks to which we add binomial, positively-truncated Gaussian, or exponential noise, as contextually described. To assess the overall quality of the reconstruction, we use the sum of root mean-squared deviations (RMSD) $\sum_{i=1}^{c} \mathrm{RMSD}(X_i, U_i\bar{V})$, and the total distance $\sum_{i=1}^{c} \|X_i - U_i\bar{V}\|$ between the decentralized data and the federated reconstructions. Additionally, we evaluate the orthogonality gap $\frac{1}{c}\sum_{i=1}^{c} \|P_i^\top P_i - \mathbb{I}\|$, which measures the D_1 alignment quality.

How precisely do different alignments estimate barycenters? We explore which alignment strategy yields the best barycenter, utilizing synthetic data of various sizes. We consider *idealized* circumstances in which each input matrix is a permutation of a synthetic ground matrix, as well as *realistic* circumstances in which each input is sampled as described earlier. Using fixed point iteration, we compare with Kuhn-Munkres (LAP, (Kuhn 1955)), Sinkhorn (SK, (Cuturi 2013)), k-nearest assignments (k-NN), as well as with k-means (KMC (Lloyd 1982)) and fuzzy c-means (FCM (Dunn 1974)). Left in Fig. 1 under *idealized* conditions, we observe that all algorithms except KMC identify the ideal barycenter for their respective alignment constraints. While most algorithms do so similarly swiftly, FCM is noticeably slower for larger sizes. Right in Fig. 1 under *realistic* condi-

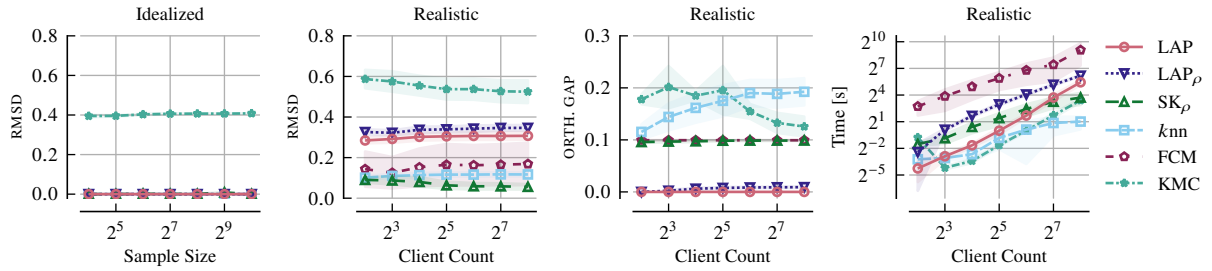

Figure 1: Our algorithms estimate barycenters efficiently and accurately. We show loss, orthogonality gap, and runtime for KMC, FCM, and fixed-point barycenters with various alignment algorithms, for increasing client count and sample sizes under idealized and realistic circumstances.

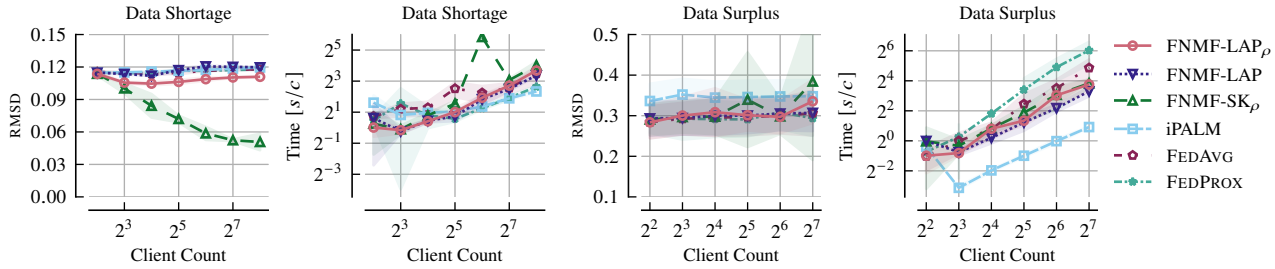

Figure 2: FNMF efficiently factorizes matrices with different client counts and data sizes accurately. We distribute a fixed amount of data (left) and proportionally-growing data (right) to an increasing number of clients, depicting runtime and RMSD.

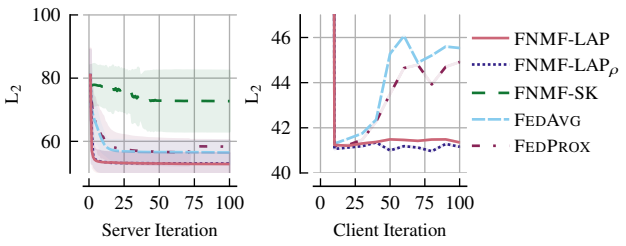

Figure 3: FNMF swiftly converges to the best-observed solution and robustly prevents drift. We show deviation with a fixed number of local steps (left), and increasing number of local steps while decreasing relative synchronizations (right).

tions, we see that strictly more powerful soft alignments, SK, SK$_\rho$, and k-NN, improve over exact methods like LAP. All methods significantly outcompete KMC in quality and FCM in speed, often by several orders of magnitude.

How does drift impact convergences? Assured that our barycenters are accurately estimated, we now study convergence and drift. Resorting to synthetic matrices, we study the convergence of FNMF. Left in Fig. 3, we fix the number of local iterations to 100 and report the loss at each synchronization, observing that while almost all methods converge quickly and smoothly, FedAvg and FedProx converge to a subpar solution and are outperformed by FNMF.

Having ensured about convergence, we explore the impact of drift, for which we increase the client optimization rounds, thereby reducing the relative amount of synchronizations. Right in Fig. 3, we see that all methods achieve about the same performance at approximately 10 rounds, after the initial drop. However, with an increasing number of rounds, we observe that FedAvg and FedProx deteriorate significantly, while FNMF stays low. Other things being equal, FNMF thus demonstrates the resilience obtained through alignments.

How well does our method scale? we analyze how error and runtime progress with increasing number of clients in two scenarios: First, for *data shortage*, we distribute a fixed number of samples to an increasing number of clients uniformly at random, producing a decreasing number of local samples. Second, for *data surplus*, we fix the data size per client, resulting in an increasing number of samples.

In Fig. 2, we observe that most algorithms robustly handle an increased client counts under surplus or under shortage of data. Our FNMF variants often outperform the similarly-performing FedProx and FedAvg, and we observe a significant gain from personalized FNMF-SK$_\rho$ when client-data are increasingly less uniform. When confronted with data shortage, SK outperforms LAP, while the latter has a consistent yet small lead under a surplus of data. Depicted in log × log-scale, we see a proportional runtime increase when federating, and significant speed-ups for the baseline iPALM, due to the lack of synchronization and decreasing subproblem sizes. Overall, alignment awareness impacts runtime only

Dataset	iPALM	MUR	FEDAVG	FEDPROX	FNMF-LAP	FNMF-LAP$_\rho$
ArXiv	–	–	0.312	0.313	0.312	**0.312**
CIFAR-10	10.652	10.634	**5.196**	5.237	5.210	5.255
Fashion MNIST	12.832	12.727	11.882	12.317	7.734	**6.733**
Goodreads	7.989	7.979	7.976	7.977	**7.942**	7.972
HPA Brain	141.603	139.623	**78.216**	81.824	83.884	86.870
Movielens	8.261	8.252	**7.702**	8.174	7.724	7.785
MNIST	12.628	12.577	11.791	12.522	6.526	**6.482**
Netflix	17.762	17.753	16.419	16.933	**16.243**	16.321
TCGA	78.116	77.645	75.858	**48.938**	84.354	83.435

Table 1: FNMF surpasses the state of the art on real-world data. We show the RMSD between nine decentralized real-world datasets and federated reconstructions using 50 clients. We underline the second lowest value, highlight the lowest value in **bold**, and designate missing data due to memory or runtime constraints by –.

insignificantly.

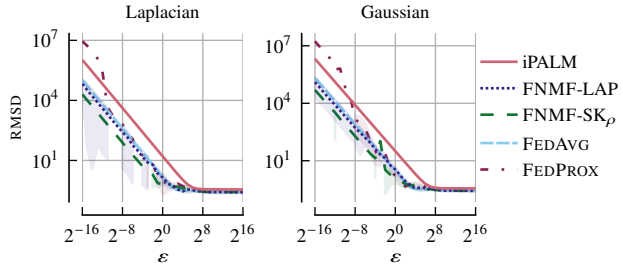

Figure 4: FNMF has state-of-the-art differentially-private performance. We show the RMSD under Laplacian and Gaussian mechanisms for varying privacy levels.

How resilient is our method under differential privacy? We evaluate performance under differential privacy using synthetic matrices. To this end, we utilize Gaussian and Laplacian mechanisms under increasing privacy levels ϵ. As shown on a $\log \times \log$-scale in Fig. 4, we see a proportional relationship between deviation and privacy level until the RMSD has converged. Notably, FNMF achieves low deviation, much earlier than iPALM, and it exhibits resilience at least on par with the differentially-private FEDPROX and FEDAVG, demonstrating its reliable privacy preservation.

Performance of FNMF on Real World Data

Having ascertained that FNMF works well on synthetic data, we explore its performance in the real world. To this end, we use nine publicly available datasets from four different domains. Covering the **recommender systems** domain, we take rating datasets *Goodreads* (Kotkov et al. 2022) (user–books), *Netflix Prize* (Netflix, Inc. 2009), and *Movielens 25M* (Harper and Konstan 2015) (user–movies). For the **biomedical** domain, we use *TCGA* (National Cancer Institute 2005) cancer gene expressions, as well as use single-cell brain protein data *HPA* from the Human Protein Atlas (Sjöstedt, Zhong, and et. al 2020). To cover **computer vision**, we use *MNIST* handwritten digits (LeCun et al. 1998), *Fashion MNIST* clothing images (Xiao, Rasul, and Vollgraf 2017), and *CIFAR-10* tiny images (Krizhevsky and Hinton 2009). In the area of **natural language processing**, we extract 200 000 random rows from the *tf-idf* matrix for all stopword-free, and lemmatized *ArXiv* abstracts (ArXiv.org Collaborations 2024).

We show the RMSD between nine decentralized datasets and federated factorizations using 50 clients for iPALM, MUR, FEDAVG, FEDPROX, and FNMF in Tab. 1 and in Apx. D. Across the board, we observe that aggregated NMF algorithms, iPALM and MUR, are outperformed by federated methods FEDPROX, FEDAVG, and FNMF. While FEDPROX's performance on *TCGA* is remarkable, it is often slightly behind FEDAVG, which particularly excels on *HPA*. Both FEDPROX and FEDAVG, however, are behind FNMF in the majority of cases. FNMF shows significant improvements, particularly with vision datasets such as *Fashion MNIST* and *MNIST*. Notably, the switch to a personalized alignment strategy with FNMF-LAP$_\rho$ results in substantial gains on *ArXiv*, *Fashion MNIST*, and *MNIST*.

5 Conclusion

We proposed a novel approach to Federated Non-Negative Matrix Factorization, tailored to distributed private matrices. We created coherence—mitigating the problem of alignment-aware drift—using a client-side proximal maps and server-side barycenters aggregation. We showed computational complexity of the barycenter problem and introduced three efficient heuristics, utilizing a wide-range of exact, incomplete, or soft alignment strategies. Combining clients and servers into our proposed FNMF algorithm, we demonstrated the effectiveness of our proposed FNMF in practice through extensive experiments on synthetic and real-world data, ensuring convergence in theory.

Future Work. Beyond the improvements introduced via FNMF, we identify several promising directions for future research aiming to advance Federated Matrix Factorization and its applications. We aim to explore novel algorithms for barycenter computations. Moreover, we see potential for using FNMF, alignment approaches, and barycenter aggregations to advance related applications, such as decentralized Deep NMF, tensor decomposition, recommender systems, and federated k-means clustering. Lastly, we aim to explore FNMF's application to privacy-sensitive life-science problems and large-scale material-science problems.

Acknowledgements

This research is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Altschuler, J. M.; and Boix-Adserà, E. 2022. Wasserstein Barycenters Are NP-Hard to Compute. *SIAM Journal on Mathematics of Data Science*, 4(1): 179–203.

Amari, S. 1999. Natural Gradient Learning for Over- and Under-Complete Bases in ICA. *Neural Computation*, 11: 1875–1883.

ArXiv.org Collaborations. 2024. ArXiv dataset and metadata of 1.7M+ scholarly papers across STEM.

Attouch, H.; Bolte, J.; and Svaiter, B. F. 2013. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.*, 137(1-2): 91–129.

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*, 394–403. PMLR.

Bolte, J.; Sabach, S.; and Teboulle, M. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494.

Choi, S. 2008. Algorithms for orthogonal nonnegative matrix factorization. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, 1828–1832. IEEE.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Neural Information Processing Systems*.

Dunn, J. C. 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1): 95–104.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, 265–284. Springer.

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.

Edmonds, J.; and Karp, R. M. 1972. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *J. ACM*, 19(2): 248–264.

Farnia, F.; Reisizadeh, A.; Pedarsani, R.; and Jadbabaie, A. 2022. An Optimal Transport Approach to Personalized Federated Learning. *IEEE Journal on Selected Areas in Information Theory*, 3: 162–171.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co. ISBN 0716710447.

Harper, F. M.; and Konstan, J. A. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4).

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 5132–5143. PMLR.

Kotkov, D.; Medlar, A.; Maslov, A.; Satyal, U. R.; Neovius, M.; and Glowacka, D. 2022. The Tag Genome Dataset for Books. In *Proceedings of the 2022 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '22)*. ISBN 978-1-4503-9186-3/22/03.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly (NRL)*, 2(1-2): 83–97.

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

Lee, D. D.; and Seung, H. S. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791.

Lee, D. D.; and Seung, H. S. 2000. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, 556–562. MIT Press.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In Dhillon, I. S.; Papailiopoulos, D. S.; and Sze, V., eds., *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org.

Li, Z.; Ding, B.; Zhang, C.; Li, N.; and Zhou, J. 2021. Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment*, 15(4).

Liu, J.; Wang, C.; Gao, J.; and Han, J. 2013. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM)*, 252–260.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137.

Luise, G.; Salzo, S.; Pontil, M.; and Ciliberto, C. 2019. Sinkhorn barycenters with free support via Frank-Wolfe algorithm. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, X. J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017,*

20-22 April 2017, Fort Lauderdale, FL, USA, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

National Cancer Institute. 2005. The Cancer Genome Atlas Program (TCGA).

Nesterov, Y. 2004. *Introductory Lectures on Convex Optimization*. Springer US. ISBN 9781441988539.

Netflix, Inc. 2009. Netflix Prize. https://archive.org/download/nf_prize_dataset.tar. Accessed: 2024-07-13.

Paatero, P.; and Tapper, U. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126.

Pock, T.; and Sabach, S. 2016. Inertial Proximal Alternating Linearized Minimization (iPALM) for Nonconvex and Nonsmooth Problems. *SIAM Journal on Imaging Sciences*, 9(4): 1756–1787.

Si, S.; Wang, J.; Zhang, R.; Su, Q.; and Xiao, J. 2022. Federated Non-negative Matrix Factorization for Short Texts Topic Modeling with Mutual Information. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, 1–7. IEEE.

Singh, A. P.; and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Knowledge Discovery and Data Mining*.

Sjöstedt, E.; Zhong, W.; and et. al, L. F. 2020. An atlas of the protein-coding genes in the human, pig, and mouse brain. *Science*, 367(6482): eaay5947.

Wang, S.; and Chang, T.-H. 2022. Federated Matrix Factorization: Algorithm Design and Application to Data Clustering. *IEEE Transactions on Signal Processing*, 70: 1625–640.

Wang, Y.; Tong, Y.; and Shi, D. 2020. Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework. In *AAAI Conference on Artificial Intelligence*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv*.

Xu, J.; Glicksberg, B. S.; Su, C.; Walker, P. B.; Bian, J.; and Wang, F. 2021. Federated Learning for Healthcare Informatics. *J. Heal. Informatics Res.*, 5(1): 1–19.

Yang, E.; Huang, Y.; Liang, F.; Pan, W.; and Ming, Z. 2021. FCMF: Federated collective matrix factorization for heterogeneous collaborative filtering. *Knowledge-Based Systems*, 220: 106946.

Yuan, X.; and Li, P. 2022. On Convergence of FedProx: Local Dissimilarity Invariant Bounds, Non-smoothness and Beyond. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.