Fast Federated Low Rank Matrix Completion

Ahmed Ali Abbasi, Shana Moothedath, and Namrata Vaswani

***Abstract*—We develop, evaluate, and analyze a communication-efficient and fast solution approach called Alternating Gradient Descent and Minimization (AltGDMin) for solving the Low Rank Matrix Completion (LRMC) problem in a federated setting. We provide detailed time and communication complexity comparisons with existing work using both order-wise complexity and numerical experiments. In addition, we provide bounds on the iteration and sample complexity needed for ϵ-accurate recovery.**

I. Introduction

We develop and analyze a fast and communication-efficient solution to low rank matrix completion (LRMC) in a federated setting. LRMC involves recovering a rank-r matrix $\boldsymbol{X}^\star \in \mathbb{R}^{n \times q}$, where $r \ll \min(n, q)$, from a subset of its entries.

A. Related Work

Starting with the seminal work of [7, 8] which introduced a nuclear norm based convex relaxation, the LRMC problem has been extensively studied in the last decade and a half [2, 6, 7, 8, 9, 10, 11, 12]. Two types of algorithms feature prominently in this literature - solutions to convex relaxations and direct iterative algorithms. The former [7, 8] are very slow: the required number of iterations for ϵ accuracy (iteration complexity) grows as $1/\sqrt{\epsilon}$ [2]. The first provably accurate iterative solution was the Alternating Minimization (AltMin) algorithm with a spectral initialization [2, 9]. AltMin was shown to converge geometrically (iteration complexity order $\log(1/\epsilon)$) with a sample complexity of $nr^{4.5}\log(1/\epsilon)$ in [2]. Subsequent work [3] considered a modified version of AltMin and improved its sample complexity to order $nr^{2.5}\log(1/\epsilon)$. Later works proposed two gradient descent (GD) based algorithms - Projected GD (ProjGD) [5, 6] and Alternating GD (AltGD) [4]. ProjGD involves GD, followed by projection onto the space of rank r matrices after each GD iteration. AltGD factorizes the unknown LR matrix $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, where $\boldsymbol{U}, \boldsymbol{B}$ have r columns and rows respectively, and updates $\boldsymbol{U}$ and $\boldsymbol{B}$ by GD. Both these methods need sample complexity of order $nr^2 \log^2 n \log^2 1/\epsilon$ samples. ProjGD converges geometrically with a constant GD step size, while AltGD needs the GD step size to be of order $1/r$. Consequently its iteration complexity is worse than that of ProjGD or AltMin by a factor of r.

To our best knowledge, there is no existing work on provably accurate federated LRMC. In tangentially related work [13, 14], distributed-computing solutions to LRMC are studied. These do not consider the federated setting, instead these assume that all data is observed centrally and then is distributed to different nodes to parallelize the computation; and these develop an approximate solution to the convex relaxation which is known to be very slow. Fully decentralized LRMC has been studied in [15, 16, 17]; these methods are slower than GD and do not come with guarantees. Other tangentially related works include [18, 19, 20], all these consider differential privacy (which is a completely different concept) or Byzantine attack resilience.

B. Contributions

The Alternating GD and Minimization (AltGDmin) algorithm was introduced in [21] as a fast solution to the LR column-wise compressive sensing problem. Our current work develops, evaluates, and analyzes AltGDMin for solving the LRMC problem in a federated setting. Sample and iteration complexity bounds are derived for it (Theorem 4.1 and Corollary 4.2). We also provide detailed time and communication complexity comparisons with existing work using both order-wise complexity and numerical experiments. We argue in Table I (details in Sec. II-A and Sec. III) that AltGDMin is private and has one of the lowest time and communication complexities.

The main focus of this letter is to compare federated AltGDmin with federated versions of other approaches in the literature that are introduced and studied for the centralized setting. To do this, we explain how a federated implementation can be developed for these approaches and what the time/communication cost would be. The sample complexity derived in our theoretical guarantee for AltGDmin is worse than what has been proved for most other iterative algorithms. We emphasize that this is an artifact of the proof techniques used in this work; see the discussion below Theorem 4.1. From simulation experiments, AltGDmin sample complexity is similar to that of any of the other iterative methods.

C. Problem Setup

LRMC involves recovering a rank-r matrix $\boldsymbol{X}^\star \in \mathbb{R}^{n \times q}$, where $r \ll \min(n, q)$, from a subset of its entries. Entry j of column k, denoted $\boldsymbol{X}^\star_{jk}$, is observed, independently of all other observations, with probability p. We use Ω to denote the set of observed indices. The observed matrix $\boldsymbol{Y} \in \mathbb{R}^{n \times q}$ is

$$\boldsymbol{Y}_{jk} = \begin{cases} \boldsymbol{X}^\star_{jk} & \text{if } (j,k) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

1) Notation: For any matrix $\boldsymbol{A}$, $\boldsymbol{a}_k$ denotes its k-th column while $\boldsymbol{a}^j$ denotes its j-th row transposed. Also, for a subset of integers $\boldsymbol{S}$, $\boldsymbol{A}_{\boldsymbol{S}}$ denotes the column sub-matrix of $\boldsymbol{A}$ with column indices in $\boldsymbol{S}$. $(\cdot)^\intercal$ denotes the matrix/vector transpose. We use $\boldsymbol{I}$ to denote the identity matrix. $\|\cdot\|$ denotes either the vector ℓ_2 norm or the induced ℓ_2 norm of a matrix while $\|\cdot\|_F$ denotes the matrix Frobenius norm. For a tall matrix $\boldsymbol{M}$, $\boldsymbol{M}^\dagger \triangleq (\boldsymbol{M}^\intercal \boldsymbol{M})^{-1}\boldsymbol{M}^\intercal$ denotes the Moore-penrose pseudo-inverse. For matrices $\boldsymbol{U}_1, \boldsymbol{U}_2$ with orthonormal columns, we

The authors are at the Department of Electrical and Computer Engineering, Iowa State University. Email: namrata@iastate.edu. A part of this work (only simulation experiments) appeared in Allerton 2023 [1].

Algorithm	Time (node) per iter.	Time (center) per iter.	Comm. (node) per iter.	Comm. (center) per iter.	Iteration (T) Complexity	Private?
AltGDmin (Proposed)	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert r^2$	nr^2	$\min(n, \sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert)r$	nr	$\log(1/\epsilon)$	yes
AltMin-Private [2, 3]	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert \max(r\log(1/\epsilon), r^2)$	0	$\min(n, \sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert)r\log(1/\epsilon)$	nr	$\log(1/\epsilon)$	yes
AltMin-Not-Private [2, 3]	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert r^2$	$\lvert\Omega\rvert r^2$	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert$	nr	$\log(1/\epsilon)$	no
AltGD [4]	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert r$	nr^2	$\min(n, \sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert)$	nr	$r\log(1/\epsilon)$	no
ProjGD [5, 6]	0	$\max(\lvert\Omega\rvert r, nr^2)\log\frac{1}{\epsilon}$	$\sum_{k\in \boldsymbol{S}_\ell} \lvert\Omega_k\rvert$	nr	$\log(\frac{n}{\epsilon})$	no

TABLE I: The Table assumes $n \approx q$, κ, μ are numerical constants (assumed in most past work on LR problems), desired accuracy level $\epsilon \leq \exp(-r)$. Observe that $|\Omega|$ needs to be greater than nr (number of degrees of freedom). Also for iterative algorithms, the best sample complexity is nr^2. Assuming total number of nodes γ to be small (treat as a numerical constant), $\sum_{k\in \boldsymbol{S}_\ell} |\Omega_k| \approx |\Omega|/\gamma$ is of order $|\Omega|$. If $|\Omega| \geq nr^2$, then $\sum_{k\in \boldsymbol{S}_\ell} |\Omega_k| \approx |\Omega| \geq nr^2$ and so $\min(n, \sum_{k\in \boldsymbol{S}_\ell} |\Omega_k|)r = nr$. Assuming $|\Omega| \geq nr^2$, AltGDmin has the smallest total node communication cost comm-cost (node) $\cdot\, T) = nr \cdot \log(1/\epsilon)$ and the smallest total time cost at the center (time (center) $\cdot\, T) = nr\log(1/\epsilon)$. It also has one of the smallest center communication costs (comm-cost (center)) $\cdot\, T = nr \cdot \log(1/\epsilon)$. Its per node total time cost (time (node)) $\cdot\, T) = |\Omega| r^2 \cdot \log(1/\epsilon)$ is larger than only that of ProjGD and comparable to that of AltMin-Not-Private and AltGD. However, for ProjGD, since all the computation needs to be done at the center, it is overall the slowest. Moreover, when compared with only the private algorithms, which is only AltMin-Private, all costs of AltGDmin are lower by a factor of $\log(1/\epsilon)$ at the nodes.

define the subspace distance between their column spans as $\mathrm{SD}_2(\boldsymbol{U}_1, \boldsymbol{U}_2) := \|(\boldsymbol{I} - \boldsymbol{U}_1\boldsymbol{U}_1^\intercal)\boldsymbol{U}_2\|$.

Let $\boldsymbol{X}^\star \overset{\mathrm{SVD}}{=} \boldsymbol{U}^\star\boldsymbol{\Sigma}^*\boldsymbol{V}^*$, where $\boldsymbol{U}^\star \in \mathbb{R}^{n\times r}$ and has orthonormal columns and $\boldsymbol{V}^\star \in \mathbb{R}^{q\times r}$ with orthonormal rows. We use κ to denote the condition number of $\boldsymbol{\Sigma}^*$. Also, we let $\boldsymbol{B}^\star := \boldsymbol{\Sigma}^*\boldsymbol{V}^*$ so that $\boldsymbol{X}^\star = \boldsymbol{U}^\star\boldsymbol{B}^\star$.

Recall that Ω is the set of indices (j, k) of observed entries of $\boldsymbol{X}^\star$. For an $n \times q$ matrix $\boldsymbol{M}$, $\boldsymbol{M}_\Omega$ is an $n \times q$ matrix in which all entries that are not in Ω are zeroed out, while those in Ω remain unchanged. We let $\Omega_k = \{(j, k) \mid (j, k) \in \Omega\}$ Also $\boldsymbol{y}_{\Omega_k} \in \mathbb{R}^{|\Omega_k|}$ denotes the sub-vector of $\boldsymbol{y}_k$ with entries in Ω_k. We define $\boldsymbol{S}_k := \boldsymbol{I}_{\Omega_k}^\intercal \in \mathbb{R}^{|\Omega_k|\times n}$ to be a row selection matrix, i.e., $\boldsymbol{S}_k\boldsymbol{M}$ selects rows of $\boldsymbol{M}$ with indices in Ω_k. Thus, $\boldsymbol{y}_{\Omega_k} = \boldsymbol{S}_k\boldsymbol{y}_k$.

2) Assumption: As in all other works on LRMC, e.g. [8], we need to the following assumption.

Assumption 1.1 (μ-incoherence of singular vectors of $\boldsymbol{X}^\star$**).** *Assume row norm bounds on $\boldsymbol{U}^\star$: $\max_{j\in[n]} \|\boldsymbol{u}^{*j}\| \leq \mu\sqrt{r/n}$, and column norm bounds on $\boldsymbol{V}^*$: $\max_{k\in[q]} \|\boldsymbol{v}_k^*\| \leq \mu\sqrt{r/q}$. Since $\boldsymbol{B}^\star = \boldsymbol{\Sigma}^*\boldsymbol{V}^*$, this implies that $\|\boldsymbol{b}_k^\star\| \leq \mu\sqrt{r/q}\sigma_{\max}^*$.*

3) Federation: Assume that there are a total of γ nodes, with $\gamma \leq q$. Each node observes entries of a subset of the columns of $\boldsymbol{Y}$ defined in (1). We use $\boldsymbol{S}_\ell$ to denote the subset of columns of $\boldsymbol{Y}$ observed by node ℓ. The sets $\boldsymbol{S}_\ell$ are mutually disjoint and $\cup_{\ell=1}^{\gamma} \boldsymbol{S}_\ell = [q]$. To keep notation simple, we assume q is a multiple of γ and $|\boldsymbol{S}_\ell| = q/\gamma$. All nodes can only communicate with a central node or "center". *In a federated setting, a desirable property is "privacy". This means that the nodes' raw data cannot be shared with the center and that the center should not be able reconstruct the matrix $\boldsymbol{X}^\star$.*

II. Alternating GD and Minimization (AltGDMin)

The goal is to minimize the squared loss cost function:

$$\min_{\check{\boldsymbol{B}}, \check{\boldsymbol{U}} :\, \check{\boldsymbol{U}}^\intercal\check{\boldsymbol{U}}=\boldsymbol{I}} f(\check{\boldsymbol{U}}, \check{\boldsymbol{B}}),\ f(\check{\boldsymbol{U}}, \check{\boldsymbol{B}}) := \|(\boldsymbol{Y} - \check{\boldsymbol{U}}\check{\boldsymbol{B}})_\Omega\|_F^2 \quad (2)$$

Imposing the orthornormal constraint on $\boldsymbol{U}$ is one way to ensure that the norm of $\boldsymbol{U}$ does not keep increasing or decreasing continuously with algorithm iterations (while that of $\boldsymbol{B}$ decreases or increases). Different from [2], which used alternating exact minimization for both $\boldsymbol{U}$ and $\boldsymbol{B}$, and different from [4], which used GD for $\boldsymbol{U}, \boldsymbol{B}$, we optimize (2) by alternating exact minimization over $\boldsymbol{B}$ and a single projected GD step for $\boldsymbol{U}$. This simple change makes our algorithm communication efficient and private, as described in Sec. II-A. The use of exact minimization for one of the variables also ensures faster error decay with iterations, so that the iteration complexity of AltGDmin is comparable to that of AltMin (see Theorem 4.1 and Table I).

As in most previous work [2, 4], we initialize $\boldsymbol{U}$ by computing the top r singular vectors of $\boldsymbol{Y}$. After this, at each algorithm iteration, we update $\boldsymbol{B}$ and $\boldsymbol{U}$ as follows.

Let $\boldsymbol{U}_k := \boldsymbol{S}_k\boldsymbol{U}$, recall that $\boldsymbol{S}_k = \boldsymbol{I}_{\Omega_k}^\intercal$. The minimization over $\boldsymbol{B}$ is a decoupled least squares (LS) problem since $f(\check{\boldsymbol{U}}, \check{\boldsymbol{B}})$ decouples as $f(\boldsymbol{U}, \check{\boldsymbol{B}}) = \sum_{k\in[q]} \|\boldsymbol{y}_{\Omega_k} - \boldsymbol{U}_k\check{\boldsymbol{b}}_k\|^2$. We update $\boldsymbol{b}_k$ as

$$\boldsymbol{b}_k = \operatorname*{argmin}_{\check{\boldsymbol{b}}} \|\boldsymbol{y}_{\Omega_k} - \boldsymbol{U}_k\check{\boldsymbol{b}}\|^2 = \boldsymbol{U}_k^\dagger\boldsymbol{y}_{\Omega_k}, \quad \text{for all } k \in [q]$$

We update $\boldsymbol{U}$ as

$$\boldsymbol{U}^+ = \mathrm{QR}(\boldsymbol{U} - \eta\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}))$$

Here $\mathrm{QR}(\boldsymbol{A})$ maps matrix $\boldsymbol{A} \in \mathbb{R}^{n\times r}$ to $\boldsymbol{Q} \in \mathbb{R}^{n\times r}$ such that $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ is the QR decomposition of $\boldsymbol{A}$. The gradient is

$$\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B}) = 2(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{Y})_\Omega\boldsymbol{B}^\intercal$$

We summarize the complete algorithm in Algorithm 1. Sample splitting (line 2) is assumed, as is common in most structured data recovery literature, e.g., [2, 5, 6].

A. Federated AltGDMin and its Complexity and Privacy

In a federated setting, at each algorithm iteration, t, each node ℓ performs two operations i) updating $\boldsymbol{b}_k$ by the LS solution, for all $k \in \boldsymbol{S}_\ell$; and ii) computation of the partial gradient $\sum_{k\in\boldsymbol{S}_\ell} [\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B})]_k = \sum_{k\in\boldsymbol{S}_\ell} (\boldsymbol{U}_k\boldsymbol{b}_k - \boldsymbol{y})_{\Omega_k}\boldsymbol{b}_k^\intercal$. Only the $n \times r$ partial gradient needs to be sent to the center. The center computes its QR decomposition and broadcasts that to all the nodes. This is used by the nodes in the next iteration. We summarize this in Algorithm 2.

Algorithm 1 AltGDMin

```
Require: partial observations $\boldsymbol{Y}$, rank $r$, step size $\eta$, and number of iterations $T$
1: Partition $\boldsymbol{Y}$ into $2T+1$ subsets $\boldsymbol{Y}_{\Omega^{(0)}}, \cdots, \boldsymbol{Y}_{\Omega^{(2T)}}$
2: Initialize $\boldsymbol{U}^{(0)}$ by top $r$ left-singular vectors of $\boldsymbol{Y}_{\Omega^{(0)}}$
3: Set all elements of $\boldsymbol{U}^{(0)}$ with magnitude greater than $2\mu\sqrt{r/n}$ to zero and orthonormalize the columns of $\boldsymbol{U}^{(0)}$
4: for $t \in 1 \cdots T$ do
5:     $\boldsymbol{b}_k^{(t)} \leftarrow (\boldsymbol{U}_k^{(t-1)})^{\dagger} \boldsymbol{y}_{\Omega_k^{(t)}}$  for all $k \in [q]$
6:     $\tilde{\boldsymbol{U}}^{(t)} \leftarrow \boldsymbol{U}^{(t-1)} - \eta(\boldsymbol{U}^{(t-1)}\boldsymbol{B}^{(t)} - \boldsymbol{Y})_{\Omega^{(T+t)}} \boldsymbol{B}^{(t)\intercal}$
7:     $\boldsymbol{U}^{(t)} \leftarrow \mathrm{QR}(\tilde{\boldsymbol{U}}^{(t)})$
8: Return $\boldsymbol{U}, \boldsymbol{B}$
```

Algorithm 2 AltGDMin-Federated

```
1: Partition $\boldsymbol{Y}$ into $2T+1$ subsets $\boldsymbol{Y}_{\Omega^{(0)}}, \cdots, \boldsymbol{Y}_{\Omega^{(2T)}}$
   //Initialize $\boldsymbol{U}^{(0)}$ by Power Method
2: Center: Initialize random $\boldsymbol{U} \in \mathbb{R}^{n \times r}$; push to nodes.
3: for $t \in 1 \cdots T_{\text{init}}$ do
4:     Node $\ell$, $\ell \in [\gamma]$
5:      $\boldsymbol{M}_\ell \leftarrow \sum_{k \in \boldsymbol{S}_\ell} \boldsymbol{y}_{\Omega_k^{(0)}} \boldsymbol{y}_{\Omega_k^{(0)}}^{\intercal} \boldsymbol{U}$; push to center.
6:     Center: $\boldsymbol{U} \leftarrow \mathrm{QR}(\sum_\ell \boldsymbol{M}_\ell)$; push to nodes.
   //AltGDMin Iterations
7: for $t \in 1 \cdots T$ do
8:     Node $\ell$, $\ell \in [\gamma]$:
9:     $\boldsymbol{b}_k^{(t)} \leftarrow (\boldsymbol{U}_k^{(t-1)})^{\dagger} \boldsymbol{y}_{\Omega_k^{(t)}}$  for all $k \in \mathcal{S}_\ell$
10:    $\mathrm{GradU}_\ell \leftarrow \sum_{k \in \boldsymbol{S}_\ell} (\boldsymbol{U}_k \boldsymbol{b}_k - \boldsymbol{y})_{\Omega_k} \boldsymbol{b}_k^{\intercal}$; push to center.
11:    Center:
12:    $\boldsymbol{U}^{(t)} \leftarrow \mathrm{QR}(\boldsymbol{U}^{(t-1)} - \eta \sum_\ell \mathrm{GradU}_\ell)$; push to nodes.
13: Return $\boldsymbol{U}, \boldsymbol{B}$
```

1) Communication Complexity of AltGDmin: The upstream (node to center) communication cost is the cost needed to transmit the partial gradient computed at node ℓ which is of size $n \times r$, but may be sparse if $\sum_{k \in \boldsymbol{S}_\ell} |\Omega_k| < n$. This will happen if $|\boldsymbol{S}_\ell| = q/\gamma$ is small, e.g., if $\gamma \approx q$. In general, this cost is $\min\left(n, \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|\right) r$. The center communicates the updated $\boldsymbol{U}$ (line 5, Algorithm 1) to the nodes with a downstream (center to node) communication complexity of nr.

2) Time Complexity of AltGDmin: Each LS problem can be solved in time $O(|\Omega_k| r^2)$ since solving an LS problem with design matrix of size $a \times b$ is $O(ab^2)$, see equation 11.14 in [22]. Gradient computation for one k, $(\boldsymbol{U}_k \boldsymbol{b}_k - \boldsymbol{y}_{\Omega_k})\boldsymbol{b}_k^{\intercal}$, costs $|\Omega_k| r$ time. Thus, the per-node time cost is $\sum_{k \in \boldsymbol{S}_\ell} \max(|\Omega_k| r, |\Omega_k| r^2) = \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k| r^2$. The QR decomposition of $\boldsymbol{U} \in \mathbb{R}^{n \times r}$ is performed at the center and costs $O(nr^2)$, see equation 10.9 in [22].

3) Privacy: The algorithm is private because, given only the partial gradient terms $\sum_{k \in \boldsymbol{S}_\ell} [\nabla_{\boldsymbol{U}} f(\boldsymbol{U}, \boldsymbol{B})]_k$ and $\boldsymbol{U}$, the center cannot recover $\boldsymbol{B}$ or the $\boldsymbol{y}_k$'s.

4) SVD Initialization: This is computed by using the power method for $\boldsymbol{Y}\boldsymbol{Y}^{\intercal}$. This costs $O(\sum_{k \in \boldsymbol{S}_\ell} |\Omega_k| r)$ per power method iteration. The upstream per-node communication cost is $O(\min(n, \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|) r)$ and the downstream cost is $O(nr)$. The method converges linearly and thus, for ϵ_{svd} accuracy, $\log(1/\epsilon_{svd})$ iterations are required. Typically, $\epsilon_{svd} = c$ or c/r suffices since SVD is only used for initialization.

III. Complexity of Benchmark Methods

A summary of the time and communication complexities derived here is provided in Table I.

1) AltMin [2]: Since LRMC measurements are both row-wise and column-wise local, the LS steps to update both $\boldsymbol{B}$ and $\boldsymbol{U}$ decouple column-wise and row-wise respectively. Hence, for LRMC, AltMin is also quite fast. In the centralized setting, both AltMin and AltGDmin have a time cost of $|\Omega| r^2$ per iteration.

When AltMin is used in a federated setting (different subsets of data observed at different nodes) *without requiring privacy*, the update of columns of $\boldsymbol{B}$ is done locally at the respective nodes, with a time cost of $\sum_{k \in \boldsymbol{S}_\ell} |\Omega_k| r^2$. These, and the observed data vectors, are transmitted to the center with a communication cost of $\sum_{k \in \boldsymbol{S}_\ell} \max(|\Omega_k|, r) = \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|$. The center implements the LS step for each row of $\boldsymbol{U}$ with a time cost of $|\Omega| r^2$ and sends the new $\boldsymbol{U}$ to the nodes. The time cost at the center, and the communication cost at the nodes, are both higher than that of AltGDmin by a factor of $|\Omega|/n$ and $|\Omega|/(\gamma n)$ respectively.

The above approach to federate AltMin does not guarantee privacy because the data needs to be sent to the center. In order to guarantee privacy, the LS step for updating $\boldsymbol{U}$ needs to be solved using multiple GD iterations. With this, the time cost at the node becomes $\sum_{k \in \boldsymbol{S}_\ell} |\Omega_k| r \max(r, \log(1/\epsilon))$, while the time cost at the center becomes zero (ignoring the cost of addition/subtraction). The communication cost from node to center now becomes $\min(n, \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|) r \log(1/\epsilon)$. Both time and communication cost are higher than those of AltGDmin by a factor of $\log(1/\epsilon)$.

2) ProjGD [5, 6]: Each iteration of Projected GD (ProjGD) involves one step of GD w.r.t. the cost function to be minimized, followed by projecting onto the constraint set which is the set of rank r matrices (by SVD) [5, 6], i.e., $\boldsymbol{X}^{+} = \mathrm{Proj}_r(\boldsymbol{X} - \eta(\boldsymbol{X} - \boldsymbol{Y})_\Omega)$. The nodes transmit the partial gradients $(\boldsymbol{x}_{\Omega_k} - \boldsymbol{y}_{\Omega_k}) \in \mathbb{R}^{|\Omega_k|}$ to the center. This has cost $\sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|$. The center computes the SVD of $(\boldsymbol{X} - \eta(\boldsymbol{X} - \boldsymbol{Y})_\Omega)$. Since $\boldsymbol{X}$ is rank-r and the second term is sparse, thus, the SVD step needs time of order $\max(n, q) r^2 + |\Omega| r$ per SVD iteration [6]. Hence the cost of this step is $\max(\max(n, q) r^2, |\Omega| r) r \log(1/\epsilon)$. The cost is dominated by the time needed for the SVD step at the T-th algorithm iteration; at this iteration the SVD accuracy needs to be of order ϵ. This method is not private because the center knows $\boldsymbol{X}$. The center then transmits $\boldsymbol{x}_{\Omega_k}$ to node k, with total downstream communication complexity $|\Omega|$.

3) Federated AltGD [4]: Alternating GD (AltGD) factorizes $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, and updates $\boldsymbol{U}$ and $\boldsymbol{B}$, followed by projecting each of them onto the set of matrices with incoherent rows and columns respectively. The GD steps are for the cost function $f(\boldsymbol{U}, \boldsymbol{B}) + \|\boldsymbol{U}^{\intercal}\boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^{\intercal}\|_F^2$. The second term is a norm balancing term that ensures the norm of $\boldsymbol{U}$ does not keep increasing with iterations while that of $\boldsymbol{B}$ decreases (or vice versa). The gradients of $f(\boldsymbol{U}, \boldsymbol{B})$ are $\nabla_{\boldsymbol{U}} = \nabla_{\boldsymbol{X}} \boldsymbol{B}^{\intercal}$ and $\nabla_{\boldsymbol{B}} = \boldsymbol{U}^{\intercal} \nabla_{\boldsymbol{X}}$ with $\nabla_{\boldsymbol{X}} = (\boldsymbol{X} - \boldsymbol{Y})_\Omega$. Parts of these can be computed at the nodes with a cost of $\min(n, \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|)$ and $\min(q, \sum_{k \in \boldsymbol{S}_\ell} |\Omega_k|)$ respectively, and sent to the center. The center just needs to concatenate or sum what is received from the nodes. The center computes the gradients of $\|\boldsymbol{U}^\intercal \boldsymbol{U} - \boldsymbol{B}\boldsymbol{B}^\intercal\|_F^2$ and performs the GD steps for both $\boldsymbol{U}, \boldsymbol{B}$. This costs $O(\max(n,q)^2 r)$. AltGD is not private because the center has access to both $\boldsymbol{U}, \boldsymbol{B}$.

IV. Sample and Iteration Complexity Guarantee

We prove the following for AltGDmin from Algorithm 1 or 2 (the latter is only a federated implementation of the former).

Theorem 4.1. *Let $\delta^{(t)} := \mathrm{SD}_2(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*)$. Assume that Assumption 1.1 holds and that, at each iteration t, entries of $\boldsymbol{X}^\star$ are observed independently with probability $p \geq C\kappa^6\mu^4 r^3/n$. Set step-size $\eta = 0.5/(p{\sigma^*_{\max}}^2)$. Given any $(C\mu\sqrt{r})$-incoherent $\boldsymbol{U}^{(0)}$ such that $\mathrm{SD}_2(\boldsymbol{U}^{(0)}, \boldsymbol{U}^*) \leq c/(\sqrt{r}\kappa^2)$, the output $\boldsymbol{U}^{(t)}$, satisfies the following: with probability (w.p.) at least $1 - 4/n^3$,*

$$\delta^{(t+1)} \leq \delta^{(t)}(1 - 0.5/\kappa^2), \tag{3}$$

and $\|\boldsymbol{X}^{(t+1)} - \boldsymbol{X}^\star\|_F \leq \delta^{(t+1)}\|\boldsymbol{X}^\star\|_F$.

We prove this result in Appendix I. Combining this with the initialization result from [2] (summarized in Lemma 5.1 in the Appendix), we can get following corollary.

Corollary 4.2. *In the setting of Theorem 4.1, if $p > C\kappa^6\mu^4 r^3(r^3 + \kappa^2 \log(\frac{1}{\epsilon}))/n$ and if $T = \kappa^2 \log(1/\epsilon)$, then, w.p. at least $1 - 4T/n^3$, the final outputs $\boldsymbol{U}^{(T)}$, $\boldsymbol{B}^{(T)}$ satisfy*

$$\mathrm{SD}_2(\boldsymbol{U}^{(T)}, \boldsymbol{U}^*) \leq \epsilon \text{ and } \|\boldsymbol{U}^{(T)}\boldsymbol{B}^{(T)} - \boldsymbol{X}^\star\|_F \leq \epsilon\|\boldsymbol{X}^\star\|_F$$

Proof. See Appendix II in ArXiv version of this work [**?**]. □

1) Discussion: Our discussion here treats κ, μ as numerical constants. The average (expected value of) the required sample complexity is nqp. AltMin [2] needs this to be $\Omega(\max(n,q)r^{4.5})$. Later work on AltMin and the works on AltGD and ProjGD need roughly $O(\max(n,q)r^2)$ samples. The reason that our result above needs a higher sample complexity is an artifact of our proof technique. We borrow the LS step bound and the initialization guarantee from [2]. The reason we need more samples than the guarantee for AltMin from [2] is because AltGDmin requires the initialization error $\delta^{(0)} \leq 1/\sqrt{r}$, while AltMin only needs $\delta^{(0)} \leq 1/2$. In ongoing work, we are working to reduce the sample complexity by (i) replacing the bound of Lemma 5.2 by our own, and (ii) replacing the clipping step in the initialization by a different approach from [4].

2) Proof Novelty: The novel ideas are in the analysis of the GD step for $\boldsymbol{U}$ and in showing that each new updated $\boldsymbol{U}$ satisfies incoherence. This needs to be treated quite differently than in case of AltGD [4] since (i) our $\boldsymbol{B}$ is not updated by GD, and (ii) since we do not use row normalization and clipping after each GD step to make $\boldsymbol{U}$ incoherent.

V. Simulation Results

MATLAB code to reproduce the results in this paper is at the first author's github repository https://github.com/aabbas02. For AltGD [4], we used the code provided by the authors of that work. For ProjGD, AltMin (Fed./NotPrvt.) and AltMin(Fed./Prvt.), we wrote our own code. As explained in Sec. II-A, the latter used GD to solve the LS problem for updating $\boldsymbol{U}$. We set the number of GD iterations for solving each LS problem to 10 and the step size to $\eta = p/\|\boldsymbol{Y}\|^2$. AltGDmin is implemented with two choices of the step size $\eta = cp/\|\boldsymbol{Y}\|^2$, $c = \{0.75, 1\}$. This approximates the step size choice suggested by Theorem 4.1 since $\mathbb{E}[\|\boldsymbol{Y}\|^2] = p{\sigma^*_{\max}}^2$. We plot the averaged subspace distance at the iteration t against the average time taken until iteration t, with the averages being computed over 100 Monte Carlo runs. The averaging is over the observed entries which are generated uniformly at random. The matrix $\boldsymbol{X}^\star = \boldsymbol{U}^\star\boldsymbol{B}^\star$ was generated once, by setting $\boldsymbol{U}^\star$ to be an orthonormalized $n \times r$ random Gaussian matrix and $\boldsymbol{B}^\star$ be an $r \times q$ random Gaussian matrix.

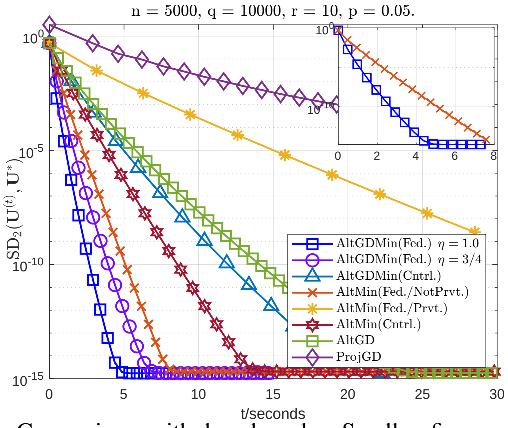

Fig. 1: Comparison with benchmarks. Smaller figure compares AltGDMin with second fastest benchmark, AltMin(Fed./NotPrvt.).

The results show that the proposed method AltGDMin is the fastest, converging to $\boldsymbol{U}^\star$ in approximately 4 seconds, compared to second fastest AltMin(Fed. NotPrvt.) which takes nearly 8 seconds.

For both AltMin (Fed./NotPrvt) and AltGDMin, we used the 'parfor' loop in MATLAB to distribute the $\boldsymbol{B}$ least-squares updates across 10 workers. AltMin (Fed./NotPrvt.) is not private because the nodes communicate the updated $\boldsymbol{b}_k^{(t+1)}$ to the center; it is slower than AltGDMin because the n $\boldsymbol{U}$-update LS problems are solved sequentially at the center with complexity $|\Omega|r^2$, compared to the nr^2 complexity of computing the QR decomposition for AltGDMin. For AltMin(Fed./Prvt.), the $\boldsymbol{U}$-update LS problems are solved by gradient descent at the nodes, as explained in Sec. III. While private, the GD version of AltMin is slow because of the communication overhead of transmitting the gradients several times in each iteration. AltMin (Cntrl.) is slow because both $\boldsymbol{U}$,$\boldsymbol{B}$ LS problems are solved sequentially by the closed form solution at the center. We did not federate the implementations of ProjGD and AltGD. The reason is in case of ProjGD all the computation needs to be done at the center. For AltGD, we computed the gradient by a matrix-matrix product at the center, which is faster than a matrix-vector product computation at the node and subsequent upstream transmission to the center. Therefore, for both algorithms, the federated implementations would have higher run-times because of the additional communication cost at each iteration.

More experiments (phase transition plots to evaluate sample complexity) are in Appendix III in ArXiv version [**?**].

References

[1] A. A. Abbasi, S. Moothedath, and N. Vaswani, “Fast federated low rank matrix completion,” in *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2023, pp. 1–6.

[2] P. Jain, P. Netrapalli, and S. Sanghavi, “Low-rank matrix completion using alternating minimization,” 2012.

[3] M. Hardt, “Understanding alternating minimization for matrix completion,” in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*. IEEE, 2014, pp. 651–660.

[4] X. Yi, D. Park, Y. Chen, and C. Caramanis, “Fast algorithms for robust pca via gradient descent,” *Advances in neural information processing systems*, vol. 29, 2016.

[5] P. Jain and P. Netrapalli, “Fast exact matrix completion with finite samples,” in *Conf. on Learning Theory*, 2015, pp. 1007–1034.

[6] Y. Cherapanamjeri, K. Gupta, and P. Jain, “Nearly-optimal robust matrix completion,” *ICML*, 2016.

[7] M. Fazel, “Matrix rank minimization with applications,” *PhD thesis, Stanford Univ*, 2002.

[8] E. J. Candes and B. Recht, “Exact matrix completion via convex optimization,” *Found. of Comput. Math*, no. 9, pp. 717–772, 2008.

[9] R. Keshavan, A. Montanari, and S. Oh, “Matrix completion from a few entries,” *IEEE Trans. Info. Th.*, vol. 56, no. 6, pp. 2980–2998, 2010.

[10] R. Sun and Z.-Q. Luo, “Guaranteed matrix completion via non-convex factorization,” *IEEE Trans. Info. Th.*, vol. 62, no. 11, pp. 6535–6579, 2016.

[11] C. Ma, K. Wang, Y. Chi, and Y. Chen, “Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution,” in *Intl. Conf. Machine Learning (ICML)*, 2018.

[12] M. Hardt and M. Wootters, “Fast matrix completion without the condition number,” in *Conf. on Learning Theory*, 2014.

[13] L. W. Mackey, A. Talwalkar, and M. I. Jordan, “Distributed matrix completion and robust factorization,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 913–960, 2015.

[14] C. Teflioudi, F. Makari, and R. Gemulla, “Distributed matrix completion,” in *2012 ieee 12th international conference on data mining*. IEEE, 2012, pp. 655–664.

[15] Q. Ling, Y. Xu, W. Yin, and Z. Wen, “Decentralized low-rank matrix completion,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2925–2928.

[16] A.-Y. Lin and Q. Ling, “Decentralized and privacy-preserving low-rank matrix completion,” *Journal of the Operations Research Society of China*, vol. 3, no. 2, pp. 189–205, 2015.

[17] M. Mardani, G. Mateos, and G. Giannakis, “Decentralized sparsity-regularized rank minimization: Algorithms and applications,” *IEEE Trans. Sig. Proc.*, 2013.

[18] Z. Li, B. Ding, C. Zhang, N. Li, and J. Zhou, “Federated matrix factorization with privacy guarantee,” *Proceedings of the VLDB Endowment*, vol. 15, no. 4, 2021.

[19] V. W. Anelli, Y. Deldjoo, T. Di Noia, A. Ferrara, and F. Narducci, “User-controlled federated matrix factorization for recommender systems,” *Journal of Intelligent Information Systems*, vol. 58, no. 2, pp. 287–309, 2022.

[20] X. He, Q. Ling, and T. Chen, “Byzantine-robust stochastic gradient descent for distributed low-rank matrix completion,” in *2019 IEEE Data Science Workshop (DSW)*. IEEE, 2019, pp. 322–326.

[21] S. Nayer and N. Vaswani, “Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections,” *IEEE Trans. Info. Th.*, Feb. 2023.

[22] L. N. Trefethen and D. Bau, *Numerical linear algebra*. Siam, 2022, vol. 181.

[23] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press, 2018, vol. 47.

Appendix I

A. Definitions

Recall that $\delta^{(t)} = \mathrm{SD}_2(\boldsymbol{U}^{(t)}, \boldsymbol{U}^*)$. Let $\boldsymbol{G} \equiv \boldsymbol{G}^{(t)} = (\boldsymbol{U}^{(t)})^\intercal \boldsymbol{X}^\star$, $\boldsymbol{U} \equiv \boldsymbol{U}^{(t)}$, $\boldsymbol{B} \equiv \boldsymbol{B}^{(t+1)}$, and $\boldsymbol{X} \equiv \boldsymbol{X}^{(t)} = \boldsymbol{U}^{(t)} \boldsymbol{B}^{(t+1)}$. Also let $\boldsymbol{U}^+ \equiv \boldsymbol{U}^{(t+1)}$. Let $\xi_{jk} \overset{\text{iid}}{\sim} Bernoulli(p)$ for all $j \in [n], k \in [q]$. Thus $\Omega_k = \{j : \xi_{jk} = 1\}$ is the set of indices of the observed entries in column k. Recall that $\boldsymbol{S}_k = \boldsymbol{I}_{\Omega_k}{}^\intercal$ and $\boldsymbol{U}_k := \boldsymbol{S}_k \boldsymbol{U}$. Let $\mu_u := C\mu\sqrt{r}$.

B. Lemmas

All lemmas below assume Assumption 1.1 (singular vectors' incoherence) holds.

Lemma 5.1. *[2, Lemma C.1 and C.2] Assume $p \geq C\kappa^6(\mu)^4 r^6/n$. Then, w.p. at least $1 - 1/n^3$, we have*

1) $\mathrm{SD}_2(\boldsymbol{U}^{(0)}, \boldsymbol{U}^*) \leq c/(\sqrt{r}\kappa^2)$.
2) $\boldsymbol{U}^{(0)}$ *is incoherent with parameter $\mu_u := C\mu\sqrt{r}$, that is, $\|\boldsymbol{u}^{j\,(0)}\| \leq \mu_u\sqrt{\frac{r}{n}}$ for all $j \in [n]$.*

Lemma 5.2. *[2, Lemma C.8] Assume $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{\frac{r}{n}}$ and $p \geq C\mu^2\mu_u^2 r^2\kappa^4/n$, $\delta^{(t)} < c/\sqrt{r}\kappa^2$. Then, w.p. greater than $1 - 1/n^3$, $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t)}\sqrt{r}\sigma^*_{\max}$.*

Lemma 5.3. *Assume $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t)}\sqrt{r}\sigma^*_{\max}$. Then,*

1) $\|\boldsymbol{X} - \boldsymbol{X}^\star\|_F \leq 2\delta^{(t)}\sqrt{r}\sigma^*_{\max}$.
2) $\|\boldsymbol{B}\| \leq (1 + \delta^{(t)}\sqrt{r})\sigma^*_{\max}$.
3) $\sigma_{\min}(\boldsymbol{B}) \geq \sqrt{1 - \delta^{(t)2}}\sigma^*_{\min} - \delta^{(t)}\sigma^*_{\max}$.

Lemma 5.4. *Assume $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$ and $p \geq C\frac{\mu_u^2 r}{n\epsilon^2}(\log n + \log r)$. Then, w.p. greater than $1 - 1/n^3$, $\|\boldsymbol{b}_k\| \leq 2\sigma^*_{\max}\mu\sqrt{\frac{r}{q}}$.*

Lemma 5.5. *Assume $\delta^{(t)} < c/\sqrt{r}\kappa^2$, $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t)}\sqrt{r}\sigma^*_{\max}$, $\|\boldsymbol{u}^j\| \leq \mu_u\sqrt{r/n}$, $\|\boldsymbol{b}_k\| \leq \sigma^*_{\max}\mu\sqrt{r/q}$, and $p \geq C\frac{\mu_u\mu}{n\epsilon^2}r(\log n + \log q)$. Then,*

1) $\|\mathrm{GradU} - \mathbb{E}[\mathrm{GradU}]\| \leq \epsilon p\sqrt{r}\delta^{(t)}\sigma^{*2}_{\min}$ *w.p. greater than* $1 - 1/n^3$,
2) $\mathbb{E}[\mathrm{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal$ *and so* $\|\mathbb{E}[\mathrm{GradU}]\| \leq p\sqrt{r}\delta^{(t)}\sigma^{*2}_{\max}$.

Lemma 5.6. *Assume $\delta^{(t)} < c/\sqrt{r}\kappa^2$, $\|\boldsymbol{B} - \boldsymbol{G}\|_F \leq \delta^{(t)}\sqrt{r}\sigma^*_{\max}$, $\|\boldsymbol{b}_k\| \leq \sigma^*_{\max}\mu\sqrt{r/q}$, and $p \geq C\mu^2 r\max(\log q, \log n)/n$. Then, w.p. greater $1 - 1/n^3$,*

1) $\|\mathrm{Grad}\boldsymbol{u}^j - \mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j]\| \leq 0.1p\|\boldsymbol{u}^j\|\sigma^{*2}_{\max}$ *and*
2) $\|\boldsymbol{u}^{j\,(t)}\| \leq (1 - 0.15/\kappa^2)\|(\boldsymbol{u}^{j\,(t-1)})\| + 0.7\|\boldsymbol{u}^{*j}\|$.

Proof. The first two lemmas are taken from [2]. The last four are proved using matrix Bernstein [23, Theorem 5.4.1] and some linear algebra tricks borrowed from [21]. The proofs are given in Appendix II in ArXiv version of this work [?]. □

C. Proof of Theorem 4.1

We prove the following claim by induction. For all times $\tau \geq 0$, (i) $\delta^{(\tau)} \leq c/\sqrt{r}\kappa^2$ and $\delta^{(\tau)} \leq (1 - c/\kappa^2)\delta^{(\tau-1)}$; (ii) $\|\boldsymbol{u}^{j\,(\tau)}\| \leq \mu_u\sqrt{r/n}$; (iii) $\|\boldsymbol{B}^{(\tau+1)} - \boldsymbol{G}^{(\tau+1)}\|_F \leq \delta^{(\tau)}\sqrt{r}\sigma^*_{\max}{}^2$; and (iv) $\|\boldsymbol{b}_k^{(\tau+1)}\| \leq 1.1\mu\sqrt{r/q}\sigma^*_{\max}$. Theorem 4.1 is the first claim of this result.

Base case: let $\delta^{(-1)} = 1$. Lemma 5.1 shows that $\delta^{(0)} \leq c/\sqrt{r}\kappa^2$ and $\|\boldsymbol{u}^{j\,(0)}\| \leq C\sqrt{r}\mu\sqrt{r/n} = \mu_u\sqrt{r/n}$. Since $c/\sqrt{r}\kappa^2 < 1/2$, this implies that $\delta^{(0)} \leq (1 - c/\kappa^2)\delta^{(-1)}$. This proves (i) and (ii) for $\tau = 0$. Lemmas 5.2 and 5.4 then prove (iii) and (iv) for $\tau = 0$.

Induction assumption: Assume the claim for $\tau = 1, 2, \ldots, t$.

Induction step: We use the last five lemmas and the induction assumption to prove the claim for $\tau = t + 1$. The induction assumption implies that Lemma 5.6 applies for all $\tau = 1, 2, \ldots, t$. Using it and the base case, $\|\boldsymbol{u}^{j\,(0)}\| \leq \mu_u\sqrt{r/n}$ and for all $\tau = 1, 2, \ldots, t$,

$$\|\boldsymbol{u}^{j\,(\tau)}\| \leq (1 - 0.15/\kappa^2)\|\boldsymbol{u}^{j\,(\tau-1)}\| + 0.7\|\boldsymbol{u}^{*j}\|$$

Applying this for each $\tau = 0, 1, \ldots, t$, $\|\boldsymbol{u}^{j\,(t)}\| \leq (1 - \frac{0.15}{\kappa^2})^t\|\boldsymbol{u}^{j\,(0)}\| + [1 + (1 - \frac{0.15}{\kappa^2}) + \cdots + (1 - \frac{0.15}{\kappa^2})^{t-1}]0.7\|\boldsymbol{u}^{*j}\| \leq (1 - \frac{0.15}{\kappa^2})^t\|\boldsymbol{u}^{j\,(0)}\| + \frac{0.7\kappa^2}{0.15}\|\boldsymbol{u}^{*j}\| \leq \|\boldsymbol{u}^{j\,(0)}\| + 5\kappa^2\|\boldsymbol{u}^{*j}\| \leq C\kappa^2\sqrt{r}\mu\sqrt{r/n} := \mu_u\sqrt{r/n}$. Thus, $\|\boldsymbol{u}^{j\,(t)}\| \leq \mu_u\sqrt{r/n}$, i.e. (ii) holds for $\tau = t$.

Using this and the induction assumption bound on $\|\boldsymbol{B} - \boldsymbol{G}\|_F$ and $\|\boldsymbol{b}_k\|$ for $\tau = t$, Lemma 5.5 applies. Using it, $\mathbb{E}[\mathrm{GradU}] = p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal$, $\|\mathbb{E}[\mathrm{GradU}]\| \leq p\sqrt{r}\delta^{(t)}\sigma^*_{\max}{}^2$, and, if $p \geq C\frac{\mu_u\mu r}{n\epsilon^2}(\log n + \log q)$, then $\|\mathrm{GradU} - \mathbb{E}\mathrm{GradU}\| \leq \epsilon p\sqrt{r}\delta^{(t)}\sigma^{*2}_{\max}$. Set $\epsilon = 0.1/\sqrt{r}\kappa^2$. Then, $\|\mathrm{GradU} - \mathbb{E}\mathrm{GradU}\| \leq 0.1p\delta^{(t)}\sigma^{*2}_{\min}$ if $p \geq C\frac{\kappa^4\mu_u\mu r^2}{n}(\log n + \log q)$, We use this to bound $\delta^{(t+1)} := \mathrm{SD}_2(\boldsymbol{U}^+, \boldsymbol{U}^*)$ is bounded as follows. Let $\boldsymbol{P} := \boldsymbol{I} - \boldsymbol{U}^\star\boldsymbol{U}^{\star\top}$, thus $\boldsymbol{P}\boldsymbol{X}^\star = \boldsymbol{0}$. Recall that $\|\boldsymbol{P}\boldsymbol{U}\| = \delta^{(t)}$, $\tilde{\boldsymbol{U}}^+ = \boldsymbol{U} - \eta\mathrm{GradU}$, and $\boldsymbol{U}^+ = \tilde{\boldsymbol{U}}^+\boldsymbol{R}^{+-1}$ where $\tilde{\boldsymbol{U}}^+ \overset{\text{QR}}{=} \boldsymbol{U}^+\boldsymbol{R}^+$. We have

$$\begin{aligned}
\delta^{(t+1)} &= \mathrm{SD}_2(\boldsymbol{U}^+, \boldsymbol{U}^\star) = \|\boldsymbol{P}\boldsymbol{U}^+\| \\
&\leq \|\boldsymbol{P}\tilde{\boldsymbol{U}}^+\| \cdot \|(\boldsymbol{R}^+)^{-1}\| = \|\boldsymbol{P}\tilde{\boldsymbol{U}}^+\|/\sigma_{\min}(\tilde{\boldsymbol{U}}^+) \\
&\leq \frac{\|\boldsymbol{P}(\boldsymbol{U} - \eta\mathbb{E}[\mathrm{GradU}] + \eta\mathbb{E}[\mathrm{GradU}] - \eta\mathrm{GradU})\|}{(1 - \eta\|\mathrm{GradU}\|)} \\
&\leq \frac{\|\boldsymbol{P}(\boldsymbol{U} - \eta p(\boldsymbol{U}\boldsymbol{B} - \boldsymbol{X}^\star)\boldsymbol{B}^\top)\| + \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU})\|}{(1 - \eta\|\mathbb{E}[\mathrm{GradU}]\| - \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU}\|)} \\
&\leq \frac{\|\boldsymbol{P}\boldsymbol{U}\| \cdot \|(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top)\| + \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU})\|}{(1 - \eta\|\mathbb{E}[\mathrm{GradU}]\| - \eta\|\mathbb{E}[\mathrm{GradU}] - \mathrm{GradU}\|)}.
\end{aligned} \tag{4}$$

Since $\delta^{(t)} \leq c/(\sqrt{r}\kappa^2)$, using the induction assumption and Lemma 5.3, $\sigma_{\min}(\boldsymbol{B}) \geq 0.9\sigma^*_{\min}$ and $\|\boldsymbol{B}\| \leq 1.1\sigma^*_{\max}$. Using these, if $\eta \leq 0.5/(p\sigma^{*2}_{\max})$, then $\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\intercal$ is positive semi-definite (psd) and $\|\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\intercal\| \leq 1 - 0.8\eta p\sigma^{*2}_{\min}$ (see details in proof of Lemma 5.6). Using this and the bounds from Lemma 5.5 in (4), and using $1/(1 - x) \leq 1 + 2x$ for $x < 0.5$,

$$\begin{aligned}
\delta^{(t+1)} &\leq \frac{\delta^{(t)}\left(1 - 0.8\eta p\sigma^{*2}_{\min} + 0.1\eta p\sigma^{*2}_{\min}\right)}{1 - \delta^{(t)}\left((1 + \epsilon)\eta p\sqrt{r}\sigma^{*2}_{\max}\right)} \\
&\leq \delta^{(t)}(1 - 0.7\eta p\sigma^{*2}_{\min})(1 + 2\delta^{(t)}((1 + \epsilon)\eta p\sqrt{r}\sigma^{*2}_{\max})) \\
&\leq \delta^{(t)}(1 - 0.7\eta p\sigma^{*2}_{\min} + \delta^{(t)} \cdot 2(1 + \epsilon)\eta p\sqrt{r}\sigma^{*2}_{\max}) \\
&\leq \delta^{(t)}(1 - \eta p\sigma^{*2}_{\min}(0.7 - \delta^{(t)}2(1 + \epsilon)\sqrt{r}\kappa^2)) \\
&\leq \delta^{(t)}(1 - \eta p\sigma^{*2}_{\min}(0.7 - 0.1))
\end{aligned}$$

The last row used $\delta^{(t)} \leq 0.1/(\sqrt{r}\kappa^2)$. Setting $\eta = 0.5/(p\sigma^{*2}_{\max})$, $\delta^{(t+1)} \leq (1 - 0.3/\kappa^2)\delta^{(t)}$. Using this and the induction assumption, $\delta^{(t+1)} \leq c/\sqrt{r}\kappa^2$. Thus claim (i) holds. Using Lemmas 5.2 and 5.4, claims (iii) and (iv) hold.

Appendix II: Proofs

D. Proof of Corollary ??

Proof. By Lemma 5.1, if $p \geq C\kappa^6(\mu)^4 r^6/n$, then $\delta^{(0)} \leq c/(\sqrt{r}\kappa^2)$ and $\boldsymbol{U}^{(0)}$ is $\mu\sqrt{r}$-incoherent. Applying the Theorem for each $t = 1, 2, \ldots, T$, we can conclude that if $p \geq C\kappa^6(\mu)^4 r^{4.5}(r^{1.5} + T)/n$, then $\mathrm{SD}_2(\boldsymbol{U}^{(T)}, \boldsymbol{U}^*) \leq (1 - 0.5/\kappa^2)^T \cdot c/(\sqrt{r}\kappa^2)$. Using the value of T, this right hand side is below ϵ. The bound on $\|\boldsymbol{X}^{(T)} - \boldsymbol{X}^\star\|_F$ then follows using Lemmas 5.2 and 5.3. □

E. Brief proof ideas

Lemma 5.3 is proved below.

Lemma 5.4 follows by writing $\boldsymbol{b}_k = (\boldsymbol{U}_k^\intercal \boldsymbol{U}_k)(\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^*)\boldsymbol{b}_k^*$ applying the Matrix-Bernstein inequality twice to bound $\|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k\|$ and $\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^*$. See section V-I.

Lemma 5.5: For 1), we use the Matrix Bernstein inequality, see section V-G. For 2), note that $\mathbb{E}[\mathrm{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\intercal$ because the expectation is taken with respect to an independent set of samples at each iteration, i.e., sample splitting.

Lemma 5.6: 1) follows by using the Matrix-Bernstein inequality. For 2), we write $\tilde{\boldsymbol{u}}^{j^{(t+1)}} = \boldsymbol{u}^{j^{(t)}} - \eta \mathrm{Grad}\boldsymbol{u}^j \pm \mathbb{E}[\mathrm{Grad}\boldsymbol{u}^j]$ and subsequently bounding $\|\boldsymbol{u}^{j^{\intercal (t+1)}}\| \leq \|\boldsymbol{R}^{-1}\| \|\tilde{\boldsymbol{u}}^{j^{(t)}}\|$. The proofs are provided in Section V-H.

F. Proof of Lemma 5.3

Writing $\boldsymbol{X}^\star = \boldsymbol{U}\boldsymbol{G} + (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal)\boldsymbol{X}^\star$, and $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{B}$, we have $\|\boldsymbol{X}^\star - \boldsymbol{X}\|_F \leq \|\boldsymbol{B} - \boldsymbol{G}\|_F + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal)\boldsymbol{U}^*\boldsymbol{B}^*\|_F \leq \|\boldsymbol{B} - \boldsymbol{G}\|_F + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\intercal)\boldsymbol{U}^*\|_F \|\boldsymbol{B}^*\| \leq \delta^{(t)}\sqrt{r}\sigma_{\max}^* + \delta^{(t)}\sqrt{r}\sigma_{\max}^*$.

For the second part, using the bound on $\|\boldsymbol{B} - \boldsymbol{G}\|$, $\|\boldsymbol{B}\| = \|\boldsymbol{B} - \boldsymbol{G} + \boldsymbol{G}\| \leq \|\boldsymbol{B} - \boldsymbol{G}\| + \|\boldsymbol{G}\| \leq \delta^{(t)}\sqrt{r}\sigma_{\max}^* + \sigma_{\max}^*$.

For the third part, $\sigma_{\min}(\boldsymbol{B}) \geq \sigma_{\min}(\boldsymbol{G}) - \sigma_{\max}(\boldsymbol{B} - \boldsymbol{G}) \geq \sqrt{1 - \delta^{(t)2}}\sigma_{\min} - \delta^{(t)}\sqrt{r}\sigma_{\max}^*$, where $\sigma_{\min}(\boldsymbol{G}) = \sigma_{\min}(\boldsymbol{U}^\intercal \boldsymbol{U}^* \boldsymbol{B}^*) \geq \sigma_{\min}(\boldsymbol{U}^\intercal \boldsymbol{U}^*)\sigma_{\min}^* \geq \sqrt{1 - \delta^{(t)2}}\sigma_{\min}^*$.

G. Proof of Lemma 5.5

The gradient with respect to $\boldsymbol{U}$ is $\mathrm{GradU} = \sum_{jk} \xi_{jk} \boldsymbol{e}_j (\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^*)\boldsymbol{b}_k^\intercal$. We will bound $\|\mathrm{GradU}\|$ by the Matrix-Bernstein inequality. Using $n \leq q$ and (5),

$$L = \max_{jk} |\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star| \max_k \|\boldsymbol{b}_k\| \leq 2\mu_u (r/\sqrt{n})\delta^{(t)}\sigma_{\max}^* \cdot \mu\sqrt{r/q}\sigma_{\max}^* \leq 2\mu_u \mu (r^{3/2}/n)\delta^{(t)}{\sigma_{\max}^*}^2,$$

where $\max_{jk} |\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^*|$ is bounded below

$$|\boldsymbol{e}_j^\top (\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{e}_k| \leq \|\boldsymbol{e}_j^\top \boldsymbol{U}\| \|\boldsymbol{B} - \boldsymbol{G}\| + \|(\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{I})\boldsymbol{U}^\star\| \|\boldsymbol{B}^\star \boldsymbol{e}_k\| \leq \mu_u \sqrt{r/n}\sqrt{r}\delta^{(t)}\sigma_{\max}^* + \mu\sqrt{r/q}\delta^{(t)}\sigma_{\max}^* \leq 2\mu_u (r/\sqrt{n})\delta^{(t)}\sigma_{\max}^*. \tag{5}$$

The variances are

$$\sigma_1^2 = p\sum_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star)^2 \boldsymbol{e}_j \boldsymbol{b}_k^\top \boldsymbol{b}_k \boldsymbol{e}_j^\top \leq p\|\boldsymbol{b}_k\|^2 \|\boldsymbol{X} - \boldsymbol{X}^\star\|_F^2 \leq p\mu^2 (r/q){\sigma_{\max}^*}^2 \cdot (\delta^{(t)}\sqrt{r}\sigma_{\max}^*)^2 = p\mu^2(r^2/q)\delta^{(t)2}{\sigma_{\max}^*}^4.$$

$$\sigma_2^2 = p\|\sum_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}_{jk}^\star)^2 \boldsymbol{e}_j^\top \boldsymbol{e}_j \boldsymbol{b}_k \boldsymbol{b}_k^\top\| \leq p\|\boldsymbol{b}_k\|^2 \|\boldsymbol{X} - \boldsymbol{X}^\star\|_F^2 = \sigma_1^2.$$

Setting $t = \epsilon p\sqrt{r}\delta^{(t)}{\sigma_{\max}^*}^2$, we have

$$\frac{t^2}{\sigma^2} = \frac{\epsilon^2 p^2 r \delta^{(t)2}{\sigma_{\max}^*}^4}{p\mu^2(r^2/q)\delta^{(t)2}{\sigma_{\max}^*}^4} = \frac{\epsilon^2 pq}{\mu^2 r} \leq \frac{\epsilon^2 pq}{\mu_u \mu r}, \quad \frac{t}{L} = \frac{\epsilon p\sqrt{r}\delta^{(t)}{\sigma_{\max}^*}^2}{\mu_u\mu(r^{3/2}/n)\delta^{(t)}{\sigma_{\max}^*}^2} = \frac{\epsilon pn}{\mu_u \mu r}.$$

By matrix Bernstein, for $\epsilon \leq 1$, w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu_u\mu r)$,

$$\|\mathrm{GradU} - \mathbb{E}[\mathrm{GradU}]\| \leq \epsilon p\sqrt{r}\delta^{(t)}{\sigma_{\max}^*}^2.$$

If $p > \mu_u \mu r \max(\log q, \log n)/n\epsilon^2$, then the above bound holds w.p. at least $1 - 1/n^3$. Also,

$$\mathbb{E}[\mathrm{GradU}] = p(\boldsymbol{X} - \boldsymbol{X}^\star)\boldsymbol{B}^\top,$$

H. Proof of Lemma 5.6

Let $\text{Grad}\boldsymbol{u}^j \in \mathbb{R}^{1\times r}$ denote the gradient of $\boldsymbol{U} \in \mathbb{R}^{n\times r}$ with respect to row j. We note the following

$$\text{Grad}\boldsymbol{u}^j = \sum_k \xi_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}^\star_{jk})\boldsymbol{b}_k^\top, \; \|\mathbb{E}[\text{GradU}_j]\| \lesssim 2p\|\boldsymbol{u}^j\|{\sigma^*_{\max}}^2.$$

$$L = \max_k |\boldsymbol{x}_{jk} - \boldsymbol{x}^\star_{jk}| \max_k \|\boldsymbol{b}_k\| \le \max(\max_k |\boldsymbol{x}_{jk}|, \max_k |\boldsymbol{x}^\star_{jk}|) \max_k \|\boldsymbol{b}_k\| \le \max(\|\boldsymbol{u}^j\|, \|\boldsymbol{u}^{*j}\|)\mu^2(r/q){\sigma^*_{\max}}^2.$$

$$\sigma_1^2 = \|\sum_k p(\boldsymbol{x}_{jk} - \boldsymbol{x}^\star_{jk})^2\boldsymbol{b}_k^\top\boldsymbol{b}_k\| \le 2p\|\sum_k {\boldsymbol{u}^j}^\top\boldsymbol{b}_k\boldsymbol{b}_k^\top\boldsymbol{u}^j\boldsymbol{b}_k^\top\boldsymbol{b}_k\| \le 2p\max_k \|\boldsymbol{b}_k\|^2{\boldsymbol{u}^j}^\top(\sum_k \boldsymbol{b}_k\boldsymbol{b}_k^\top)\boldsymbol{u}^j \le 2p\|\boldsymbol{u}^j\|^2\mu^2(r/q){\sigma^*_{\max}}^4.$$

$$\sigma_2^2 = \|\sum_k p(\boldsymbol{x}_{jk} - \boldsymbol{x}^\star_{jk})^2\boldsymbol{b}_k\boldsymbol{b}_k^\top\| \le 2p\|\sum_k {\boldsymbol{u}^j}^\top\boldsymbol{b}_k\boldsymbol{b}_k^\top\boldsymbol{u}^j\boldsymbol{b}_k\boldsymbol{b}_k^\top\| \le 2p\max_k \|\boldsymbol{b}_k\|^2\|{\boldsymbol{u}^j}^\top(\sum_k \boldsymbol{b}_k\boldsymbol{b}_k^\top)\boldsymbol{u}^j\| \le 2p\|\boldsymbol{u}^j\|^2\mu^2(r/q){\sigma^*_{\max}}^4.$$

Here, $\sigma_1^2 \equiv \mathbb{E}[\sum_k \xi_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}^*_{jk})^2\boldsymbol{b}_k^\intercal\boldsymbol{b}_k]$ and $\sigma_2^2 \equiv \mathbb{E}[\sum_k \xi_{jk}(\boldsymbol{x}_{jk} - \boldsymbol{x}^*_{jk})^2\boldsymbol{b}_k\boldsymbol{b}_k^\intercal]$. By the matrix Bernstein inequality with $t = \epsilon p\|\boldsymbol{u}^j\|{\sigma^*_{\min}}^2$, we have w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu^2\kappa^4 r)$,

$$\|\mathbb{E}[\text{GradU}_j] - \text{GradU}_j\| \le \epsilon p\|\boldsymbol{u}^j\|{\sigma^*_{\min}}^2.$$

This completes the proof for the first part of the lemma.

By line 5 of Algorithm 1, adding/subtracting $\mathbb{E}[\text{GradU}_j] = p{\boldsymbol{u}^{*j}}^\top\boldsymbol{B}^\star\boldsymbol{B}^\top$, and using the above bound on $\|\mathbb{E}[\text{GradU}_j] - \text{GradU}_j\|$

$$\tilde{\boldsymbol{u}}_j^{(t+1)^\top} = {\boldsymbol{u}^j}^{(t)\top}(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top) - \eta\text{GradU}_j = {\boldsymbol{u}^j}^{(t)\top}(\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top) - \eta p{\boldsymbol{u}^{*j}}^\top\boldsymbol{B}^\star\boldsymbol{B}^\top + \eta(\mathbb{E}[\text{GradU}_j] - \text{GradU}_j),$$

w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu^2\kappa^4 r)$.

Using the assumed bound on $\boldsymbol{B} - \boldsymbol{G}$ and Lemma 5.3 and $\delta^{(t)} \le c/\sqrt{r}\kappa^2$, $\sigma_{\min}(\boldsymbol{B}) \ge 0.9\sigma^*_{\min}$ and $\sigma_{\max}(\boldsymbol{B}) \le 1.1\sigma^*_{\max}$. Thus, if $\eta < 0.5/p{\sigma^*_{\max}}^2$ then, $\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top$ is positive semi-definite (psd) and $\|\boldsymbol{I} - \eta p\boldsymbol{B}\boldsymbol{B}^\top\| = 1 - \eta p\sigma^2_{\min}(\boldsymbol{B}) \le 1 - 0.9\eta p{\sigma^*_{\min}}^2$. Thus, if $\eta < 0.5/p{\sigma^*_{\max}}^2$, then,

$$\|\tilde{\boldsymbol{u}}_j^{(t+1)}\| \le \|{\boldsymbol{u}^j}^{(t)}\|(1 - 0.9\eta p{\sigma^*_{\min}}^2) + \epsilon\eta p{\sigma^*_{\min}}^2\|{\boldsymbol{u}^j}^{(t)}\| + 1.1\eta p\|\boldsymbol{u}^{*j}\|{\sigma^*_{\max}}^2 \le (1 - (0.9 - \epsilon)\eta p{\sigma^*_{\min}}^2)\|{\boldsymbol{u}^j}^{(t)}\| + \eta p{\sigma^*_{\max}}^2\|\boldsymbol{u}^{*j}\|. \tag{6}$$

w.p. at least $1 - \exp(\log q - \epsilon^2 pn/\mu^2\kappa^4 r)$.

We bound $\|{\boldsymbol{u}^j}^{\intercal(t+1)}\| \le \|(\boldsymbol{R}^{(t+1)})^{-1}\| \cdot \|\tilde{\boldsymbol{u}}_j^{\intercal(t+1)}\|$, where $\tilde{\boldsymbol{U}}^{(t+1)} \overset{\text{QR}}{=} \boldsymbol{U}^{(t+1)}\boldsymbol{R}^{(t+1)}$, and

$$\|(\boldsymbol{R}^{(t+1)})^{-1}\| = 1/(\sigma_{\min}(\boldsymbol{U} - \eta\|\text{GradU}\|) \le 1/(1 - \eta p(1+\epsilon)\sqrt{r}\delta^{(t)}\sigma^{*2}_{\max}) \le 1/(1 - 0.25\eta p{\sigma^*_{\min}}^2) \le 1 + 0.5\eta p{\sigma^*_{\min}}^2. \tag{7}$$

w.p. given in Lemma 5.5. In the above we have used Lemma 5.5 and the upper bound on $\delta^{(t)}$. Thus,

$$\begin{aligned}
\|\boldsymbol{u}_j^{(t+1)}\| &\le (1 + 0.5\eta p{\sigma^*_{\min}}^2)(1 - (0.9 - \epsilon)\eta p{\sigma^*_{\min}}^2)\|{\boldsymbol{u}^j}^{(t)}\| + (1 + 0.5\eta p{\sigma^*_{\min}}^2)\eta p{\sigma^*_{\max}}^2\|\boldsymbol{u}^{*j}\| \\
&\le (1 - (0.4 - \epsilon)\eta p{\sigma^*_{\min}}^2)\|{\boldsymbol{u}^j}^{(t)}\| + (1 + 0.5\eta p{\sigma^*_{\min}}^2)\eta p{\sigma^*_{\max}}^2\|\boldsymbol{u}^{*j}\| \\
&\le (1 - (0.4 - \epsilon)\eta p{\sigma^*_{\min}}^2)\|{\boldsymbol{u}^j}^{(t)}\| + (1 + 0.25/\kappa^2)0.5\|\boldsymbol{u}^{*j}\| \\
&\le (1 - \frac{0.15}{\kappa^2})\|{\boldsymbol{u}^j}^{(t)}\| + 0.7\|\boldsymbol{u}^{*j}\|
\end{aligned}$$

where the last bound follows by setting $\eta = 0.5/p{\sigma^*_{\max}}^2$, $\epsilon = 0.1$.

Thus, we have shown that if $\delta^{(t)} \le c/\sqrt{r}\kappa^2$, and $\eta = 0.5/p{\sigma^*_{\max}}^2$, w.p. at least $1 - 3/n^3$,

$$\|{\boldsymbol{u}^j}^{(t+1)}\| \le (1 - \frac{0.15}{\kappa^2})\|{\boldsymbol{u}^j}^{(t)}\| + 0.7\|\boldsymbol{u}^{*j}\|$$

if p satisfies the stated bound in the lemma.

I. Proof of Lemma 5.4

We have

$$\boldsymbol{b}_k = \underbrace{(\boldsymbol{U}_k^\intercal\boldsymbol{U}_k)^{-1}}_{T_1}\underbrace{\boldsymbol{U}_k^\intercal\boldsymbol{U}_k^*}_{T_2}\boldsymbol{b}_k^*. \tag{8}$$

The following has been proved in Lemma C.6 of [2]. If $p \ge C\frac{\mu_u^2 r}{n\epsilon^2}(\log n + \log r)$, then with probability greater than $1 - \frac{1}{n^3}$,

$$\|(\boldsymbol{U}_k^\intercal\boldsymbol{U}_k)^{-1}\| \le \frac{1}{(1-\epsilon)p}. \tag{9}$$

Consider the second term, T_2. The term $\|T_2 - \mathbb{E}[T_2]\|$ can be bounded by the Matrix-Bernstein inequality. We have $\mathbb{E}[T_2] = p\boldsymbol{U}^\intercal \boldsymbol{U}^* \neq p\boldsymbol{I}$. By the reverse triangle inequality,

$$\|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^* - p\boldsymbol{U}^\intercal \boldsymbol{U}^*\| \geq \|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^*\| - p, \tag{10}$$

where we have used $\|\boldsymbol{U}^\intercal \boldsymbol{U}^*\| \leq \|\boldsymbol{U}\|\|\boldsymbol{U}^*\| \leq 1$. Therefore,

$$\|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^*\| \leq \|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^* - p\boldsymbol{U}^\intercal \boldsymbol{U}^*\|_{op} + p. \tag{11}$$

The second difference from the proof of (9) is that $T_2 - \mathbb{E}[T_2]$ is not symmetric. The variance σ^2 is now

$$\sigma^2 = \max(\|\sum_j \mathbb{E}[\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal]\|, \|\sum_j \mathbb{E}[\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j]\|),$$

where $\boldsymbol{Z}_j = (\xi_{jk} - p)\boldsymbol{u}^j \boldsymbol{u}^{j*\intercal}$, and

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{Z}_j] = 0,\\
&\|\boldsymbol{Z}_j\| = \max(1-p, p)\|\boldsymbol{u}^j\|\|\boldsymbol{u}^{j*}\| \leq \mu_u \mu \frac{r}{n},\\
&\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal \sim \begin{cases} (1-p)^2\|\boldsymbol{u}^{j*}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal} & \text{w.p. } p\\ p^2\|\boldsymbol{u}^{j*}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal} & \text{w.p. } 1-p, \end{cases}\\
&\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j \sim \begin{cases} (1-p)^2\|\boldsymbol{u}^{j}\|^2 \boldsymbol{u}^{j*} \boldsymbol{u}^{j*\intercal} & \text{w.p. } p\\ p^2\|\boldsymbol{u}^{j}\|^2 \boldsymbol{u}^{j*} \boldsymbol{u}^{j*\intercal} & \text{w.p. } 1-p, \end{cases}
\end{aligned}$$

Both variances $\sigma^2(\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal)$ and $\sigma^2(\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j)$ can be bounded by the same upper bound. Note that

$$\begin{aligned}
\mathbb{E}[\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal] &= p(1-p)^2\|\boldsymbol{u}^{*j}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal} + p^2(1-p)\|\boldsymbol{u}^{*j}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal}\\
&\triangleq p'\|\boldsymbol{u}^{j*}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal}, \quad \text{where } p' = p(1-p)^2 + p^2(1-p) \leq 2p.
\end{aligned} \tag{12}$$

The variance $\sigma^2(\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal)$ is

$$\begin{aligned}
\sigma^2(\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal) = \|\sum_{j=1}^{j=n} \mathbb{E}[\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal]\| &= p'\|\sum_{j=1}^{j=n} \|\boldsymbol{u}^{j*}\|^2 (\boldsymbol{u}^j \boldsymbol{u}^{j\intercal})\|\\
&= p' \max_{\boldsymbol{w}:\, \|\boldsymbol{w}\|=1} \sum_{j=1}^{j=n} \boldsymbol{w}^\intercal (\|\boldsymbol{u}^{j*}\|^2 \boldsymbol{u}^j \boldsymbol{u}^{j\intercal})\boldsymbol{w}\\
&\leq p'\mu^2 \frac{r}{n} \max_{\boldsymbol{w}:\, \|\boldsymbol{w}\|=1} \boldsymbol{w}^\intercal \sum_{j=1}^{j=n} \boldsymbol{u}^j \boldsymbol{u}^{j\intercal} \boldsymbol{w}\\
&\leq p'\mu^2 \frac{r}{n} \max_{\boldsymbol{w}:\, \|\boldsymbol{w}\|=1} \boldsymbol{w}^\intercal \boldsymbol{U}^\intercal \boldsymbol{U} \boldsymbol{w}\\
&= p'\mu^2 \frac{r}{n} \max_{\boldsymbol{w}} \boldsymbol{w}^\intercal \boldsymbol{w}\\
&= p'\mu^2 \frac{r}{n} \leq 2p\mu^2 \frac{r}{n}.
\end{aligned}$$

$\sigma^2(\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j)$ can be similarly bounded as $\sigma^2(\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j) \leq 2p\mu_u^2 \frac{r}{n}$. Because $\mu_u \geq \mu$, we have $\sigma^2(\boldsymbol{Z}_j \boldsymbol{Z}_j^\intercal) \geq \sigma^2(\boldsymbol{Z}_j^\intercal \boldsymbol{Z}_j)$. By the Matrix-Bernstein inequality, for $\epsilon \leq 2$ and w.p. greater than $1 - \exp(\log 2r - c\frac{n\epsilon^2 p}{\mu_u^2 r})$

$$\|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^* - p\boldsymbol{U}^\intercal \boldsymbol{U}^*\| \leq \epsilon p. \tag{13}$$

From (11) and (13), w.h.p,

$$\|\boldsymbol{U}_k^\intercal \boldsymbol{U}_k^*\| \leq (1+\epsilon)p. \tag{14}$$

Substituting (9), (14) in (8), and setting $\epsilon = 1/10$, we have

$$\|\boldsymbol{b}_k\| \leq \frac{3}{2}\|\boldsymbol{b}_k^*\| \leq \frac{3}{2}\sigma_{\max}^* \mu \sqrt{r/q}. \tag{15}$$