Optimal Analysis of Subset-Selection Based ℓ_p Low-Rank Approximation

Chen Dan
Carnegie Mellon University
cdan@cs.cmu.edu

Hong Wang*
Massachusetts Institute of Technology
hongwang@mit.edu

Hongyang Zhang*
Carnegie Mellon University
honyanz@cs.cmu.edu

Yuchen Zhou*
University of Wisconsin, Madison
yuchenzhou@stat.wisc.edu

Pradeep Ravikumar
Carnegie Mellon University
pradeepr@cs.cmu.edu

Abstract

We study the low rank approximation problem of any given matrix A over $\mathbb{R}^{n\times m}$ and $\mathbb{C}^{n\times m}$ in entry-wise ℓ_p loss, that is, finding a rank-k matrix X such that $\|A-X\|_p$ is minimized. Unlike the traditional ℓ_2 setting, this particular variant is NP-Hard. We show that the algorithm of column subset selection, which was an algorithmic foundation of many existing algorithms, enjoys approximation ratio $(k+1)^{1/p}$ for $1 \le p \le 2$ and $(k+1)^{1-1/p}$ for $p \ge 2$. This improves upon the previous $O(k+1)$ bound for $p \ge 1$. We complement our analysis with lower bounds; these bounds match our upper bounds up to constant 1 when $p \ge 2$. At the core of our techniques is an application of *Riesz-Thorin interpolation theorem* from harmonic analysis, which might be of independent interest to other algorithmic designs and analysis more broadly.

Our analysis results in improvements on approximation guarantees of several other algorithms with various time complexity. For example, to make the algorithm of column subset selection computationally efficient, we analyze a polynomial time bi-criteria algorithm which selects $O(k \log m)$ number of columns. We show that this algorithm has an approximation ratio of $O((k+1)^{1/p})$ for $1 \le p \le 2$ and $O((k+1)^{1-1/p})$ for $p \ge 2$. This improves over the best-known bound with an $O(k+1)$ approximation ratio. Our bi-criteria algorithm also implies an exact-rank method in polynomial time with a slightly larger approximation ratio.

1 Introduction

Low rank approximation has wide applications in compressed sensing, numerical linear algebra, machine learning, and many other domains. In compressed sensing, low rank approximation serves as an indispensable building block for data compression. In numerical linear algebra and machine learning, low rank approximation is the foundation of many data processing algorithms, such as PCA. Given a data matrix $A \in \mathbb{F}^{n\times m}$, low rank approximation aims at finding a low-rank matrix $X \in \mathbb{F}^{n\times m}$ such that

$$\mathrm{OPT} = \min_{X:\mathrm{rank}(X)\le k} \|X-A\|. \tag{1}$$

Preprint. Under review.

Here the field $\mathbb{F}$ can be both $\mathbb{R}$ and $\mathbb{C}$. The focus of this work is on the case when $\|\cdot\|$ is the ℓ_p norm, and we are interested in an estimate $\widehat{X}$ with a *tight* approximation ratio α so that we have the guarantee: $\|\widehat{X} - A\| \leq \alpha \cdot \mathsf{OPT}$,

As noted earlier, such low-rank approximation is a fundamental workhorse of machine learning. The key reason to focus on approximations with respect to general ℓ_p norms, in contrast to the typical ℓ_2 norm, is that these general ℓ_p norms are better able to capture a broader range of realistic noise in complex datasets. For example, it is well-known that the ℓ_1 norm is more robust to the sparse outlier [1–3]. So the ℓ_1 low-rank approximation problem is a robust version of the classic PCA which uses the ℓ_2 norm and has received tremendous attentions in machine learning, computer vision and data mining [4], [5], [6]. While it has better empirical and statistical properties, the key bottleneck to solving the problem in (1) is computational, and is known to be NP-hard in general. For example, the ℓ_1 low-rank approximation is NP-hard to solve exactly even when $k = 1$ [7], and is even hard to approximate with large error under the Exponential Time Hypothesis [8]. [9] proved the NP-hardness of the problem when $p = \infty$. A recent work [10] proves that the problem has no constant factor approximation algorithm running in time $O(2^{k^\delta})$ for a constant $\delta > 0$, assuming the correctness of Small Set Expansion Hypothesis and Exponential Time Hypothesis. The authors also proposed a PTAS (Polynomial Time Approximation Scheme) with $(1 + \varepsilon)$ approximation ratio when $0 < p < 2$. However, the running time is as large as $O(n^{\mathsf{poly}(k/\varepsilon)})$.

Many other efforts have been devoted to designing approximation algorithms in order to alleviate the computational issues of ℓ_p low-rank approximation. One promising approach is to apply subgradient descent based methods or alternating minimization [11]. Unfortunately, the loss surface of problem (1) suffers from spurious local minima and saddle points [12], which might be arbitrarily worse than OPT. Therefore, they may not work well for the low-rank approximation problem as these local searching algorithms may easily get stuck at bad stationary points without any guarantee.

Instead, we consider another line of research—the heuristic algorithm of column subset selection (CSS). Here, the algorithm proceeds by choosing the best k columns of A as an estimation of column space of X and then solving an ℓ_p linear regression problem in order to obtain the optimal row space of X. See Algorithm 1 for the detailed procedure. Although the vanilla form of the subset selection based algorithm also has an exponential time complexity in terms of the rank k, it can be slightly modified to polynomial time bi-criteria algorithms which selects more than k columns [13]. Most importantly, these algorithms are easy to implement and runs fast with nice empirical performance. Thus, subset selection based algorithms might seem to effectively alleviate the computational issues of problem (1). The caveat however is that CSS might seem like a simple heuristic, with potentially a very large worst-case approximation ratio α.

In this paper, we show that CSS yields surprisingly reasonable approximation ratios, which we also show to be tight by providing corresponding lower bounds, thus providing a strong theoretical backing for the empirical observations underlying CSS.

Due in part to its importance, there has been a burgeoning set of recent analyses of column subset selection. In the traditional low rank approximation problem with Frobenious norm error (the $p = 2$ case in our setting), [14] showed that CSS achieves $\sqrt{k+1}$ approximation ratio. The authors also showed that the $\sqrt{k+1}$ bound is tight (both upper and lower bounds can be recovered by our analysis). [15–18] improved the running time of CSS with different sampling schemes while preserving similar approximation bounds. The CSS algorithm and its variants are also applied and analyzed under various different settings. For instance, [19] and [20] studied the CUR decomposition with the Frobenius norm. [21] studied the CSS problem under the missing-data case. With ℓ_1 error, [22] studied CSS for non-negative matrices in ℓ_1 error. [23] gave tight approximation bounds for CSS under finite-field binary matrix setting. Furthermore, [8] considered the low rank tensor approximation with the Frobenius norm.

Despite a large amount of work on the subset-selection algorithm and the ℓ_p low rank approximation problem, many fundamental questions remain unresolved. Probably one of the most important open questions is: what is the *tight* approximation ratio α for the subset-selection algorithm in the ℓ_p low rank approximation problem, up to a constant factor? In [13], the approximation ratio is shown to be upper bounded by $(k + 1)$ and lower bounded by $O(k^{1-\frac{2}{p}})$ when $p > 2$. This problem becomes even more challenging when one requires the approximation ratio to be tight up to factor 1, as little was known about a direct tool to achieve this goal in general. In this work, we improve both upper and

Algorithm 1 A $c_{p,k}$ approximation to problem (1) by column subset selection.

```
Input: Data matrix $A$ and rank parameter $k$.
Output: $X \in \mathbb{R}^{n\times m}$ such that $\mathrm{rank}(X) \le k$ and $\|X - A\|_p \le c_{p,k} \cdot \mathsf{OPT}$.
for $I \in \{S \subseteq [m];\ |S| = k\}$ do
    $U \leftarrow A_I$.
    Run $\ell_p$ linear regression over $V$ that minimizes the loss $\|A - UV\|_p$.
    Let $X_I = UV$
end for
return $X_I$ which minimizes $\|A - X_I\|_p$ for $I \in \{S \subseteq [m];\ |S| = k\}$.
```

lower bounds in [13] to optimal when $p > 2$. Note that our bounds are still applicable and improve over [13] when $1 < p < 2$, but there is an $O(k^{\frac{2}{p}-1})$ gap between the upper and lower bounds.

1.1 Our Results

The best-known approximation ratio of subset selection based algorithms for ℓ_p low-rank approximation is $O(k + 1)$ [13]. In this work, we give an improved analysis of this algorithm. In particular, we show that the Column Subset Selection in Algorithm 1 is a $c_{p,k}$-approximation, where

$$c_{p,k} = \begin{cases} (k+1)^{\frac{1}{p}}, & 1 \le p \le 2, \\ (k+1)^{1-\frac{1}{p}}, & p \ge 2. \end{cases}$$

This improves over Theorem 4 in [13] which proved that the algorithm is an $O(k+1)$-approximation, for all $p \ge 1$. Below, we state our main theorem formally:

Theorem 1.1 (Upper bound). *The subset selection algorithm in Algorithm 1 is a $c_{p,k}$-approximation.*

Our proof of Theorem 1.1 is built upon novel techniques of Riesz-Thorin interpolation theorem. In particular, with the proof of special cases for $p = 1, 2, \infty$, we are able to interpolate the approximation ratio of all intermediate p's. Our techniques might be of independent interest to other ℓ_p norm or Schatten-p norm related problem more broadly. See Section 1.2 for more discussions.

We also complement our positive result of subset selection algorithm with a negative result. Surprisingly, our upper bound matches our lower bound exactly up to constant 1 for $p \ge 2$. Below, we state our negative results formally:

Theorem 1.2 (Lower bound). *There exist infinitely many different values of k, such that approximation ratio of any k-subset-selection based algorithm is at least $(k+1)^{1-\frac{1}{p}}$ for ℓ_p rank-k approximation.*

Note that the lower bound strictly improves the result in [13], which is $(k+1)^{1-\frac{2}{p}}$. The main idea of the proof can be found in Section 1.2 and we put the whole proof of Theorem 1.2 in Appendix 3.

Algorithm 1 requires exponential time to solve. However, it serves as an algorithmic foundation of many existing computationally efficient algorithms. For example, a bi-criteria variant of this algorithm runs in polynomial time, only requiring the rank parameter to be a little over-parameterized. Our new analysis can be applied to this algorithm as well. Below, we state our result informally:

Theorem 1.3 (Informal statement of Theorem 4.1). *There is a bi-criteria algorithm which runs in $\mathsf{poly}(m, n, k)$ time and selects $k \log m$ columns of A. The algorithm is an $O(c_{p,k})$-approximation algorithm.*

Our next result is a computationally-efficient, *exact-rank* algorithm with slightly larger approximation ratio. Below, we state our result informally:

Theorem 1.4 (Informal statement of Theorem 4.2). *There is an algorithm which solves problem (1) and runs in $\mathsf{poly}(m, n)$ time with an $O(c_{p,k}^3 k \log m)$-approximation ratio, provided that $k = O(\frac{\log n}{\log \log n})$.*

1.2 Our Techniques

In this section, we give a detailed discussion about our techniques in the proofs. We start with the analysis of approximation ratio of *column subset selection* algorithm.

Remark Throughout this paper, we state the theorems for real matrices. The results can be naturally generalized for complex matrices as well.

Notations: We denote by $A \in \mathbb{R}^{n \times m}$ the input matrix, and A_i the i-th column of A. $A^* = UV$ is the optimal rank-k approximation, where $U \in \mathbb{R}^{n \times k}, V \in \mathbb{R}^{k \times m}$. $\Delta_i = A_i - UV_i$ is the error vector on the i-th column, and Δ_{li} the l-th element of vector Δ_i. For any $X \in \mathbb{R}^{n \times k}$, define the error of projecting A onto X by $\mathrm{Err}(X) = \min_{Y \in \mathbb{R}^{k \times m}} \|A - XY\|_p$. Let $J = (j_1, \cdots, j_k) \in [m]^k$ be a subset of $[m]$ with cardinality k. We denote X_J as the following column subset in matrix X: $X_J = [X_{j_1}, X_{j_2}, \cdots, X_{j_k}]$. Similarly, we denote by X_{dJ} the following column subset in the d-th dimension of matrix X: $X_{dJ} = [X_{dj_1}, X_{dj_2}, \cdots, X_{dj_k}]$. Denote by J^* the column subset which gives smallest approximation error, i.e., $J^* = \mathrm{argmin}_{J \in [m]^k} \mathrm{Err}(A_J)$.

Analysis in Previous Work: In order to show that the column subset selection algorithm gives an α-approximation, we need to prove that

$$\mathrm{Err}(A_{J^*}) \leq \alpha \|\Delta\|_p. \tag{2}$$

Directly bounding $\mathrm{Err}(A_{J^*})$ is prohibitive. In [13], the authors proved an upper bound of $(k+1)$ in two steps. First, the authors constructed a specific $S \in [m]^k$, and upper bounded $\mathrm{Err}(A_{J^*})$ by $\mathrm{Err}(A_S)$. Their construction is as follows: S is defined as the minimizer of

$$S = \operatorname*{argmin}_{J \in [m]^k} \frac{|\det(V_J)|}{\prod_{j \in J} \|\Delta_j\|_p}.$$

In the second step, [13] upper bounded $\mathrm{Err}(A_S)$ by considering the approximation error on each column A_i, and upper bounded the ℓ_p distance from A_i to the subspace spanned by A_S using triangle inequality of ℓ_p distance. They showed that the distance is at most $(k+1)$ times of $\|\Delta_i\|$, uniformly for all columns $i \in [m]$. Therefore, the approximation ratio is bounded by $(k+1)$.Our approach is different from the above analysis in both steps.

Weighted Average: In the first step, we use a so-called *weighted average* technique, inspired by the approach in [23]. Instead of using the error of one specific column subset as an upper bound, we use a weighted average over all possible column subsets, i.e.,

$$\mathrm{Err}^p(A_{J^*}) \leq \sum_{J \in [m]^k} w_J \mathrm{Err}^p(A_J),$$

where the weight w_J's are carefully chosen for each column subset J. This weighted average technique captures more information from all possible column subsets, rather than only from one specific subset, and leads to a tighter bound.

Riesz-Thorin Interpolation Theorem: In the second step, unlike [13] which simply used triangle inequality to prove the upper bound, our technique leads to more refined analysis of upper bounds for the approximation error for each subset $J \in [m]^k$. With the technique of weighted average in the first step, proving a technical inequality (Lemma 2.2) concerning the determinants suffices to complete the analysis of approximation ratio. In the proof of this lemma, we introduce several powerful tools from harmonic analysis, the theory of interpolating linear operators. Riesz-Thorin theorem is a classical result in interpolation theory that gives bounds for L^p to L^q operator norm. In general, it is easier to prove estimates within spaces like L^2, L^1 and L^∞. Interpolation theory enables us to generalize results in those spaces to some L^p and L^q spaces with an explicit operator norm. By the Riesz-Thorin interpolation theorem, we are able to prove the lemma by just checking the special cases $p = 1, 2, \infty$, and then interpolate the inequality for all the intermediate value of p's.

Lower Bounds: We now discuss the techniques in proving the lower bounds. Our proof is a generalization of [14], which shows that for the special case $p = 2$, $\sqrt{k+1}$ is the best possible approximation ratio. Their proof for the lower bound is constructive: they constructed a $(k+1) \times$

$(k+1)$ matrix A, such that using any k-subset leads to a sub-optimal solution by a factor no less than $(1-\varepsilon)\sqrt{k+1}$. However, since ℓ_p norm is not rotationally-invariant in general, it is tricky to generalize their analysis to other values of p's. To resolve the problem, we use a specialized version of their construction, the perturbed Hadamard matrices (see Section 3 for details), as they have nice symmetricity and are much easier to analyze. We give an example of special case $k=3$ for better intuition:

$$A = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Here ε is a positive constant close to 0. We note that A is very close to a rank-3 matrix: if we replace the first row by four zeros, then it becomes rank-3. Thus, the optimal rank-3 approximation error is at most $(4\varepsilon^p)^{1/p} = 4^{1/p}\varepsilon$. Now we consider the column subset selection algorithm. For example, we use the first three columns A_1, A_2, A_3 to approximate the whole matrix — the error only comes from the fourth column. We can show that when ε is small, the projection of A_4 to $\mathrm{span}(A_1, A_2, A_3)$ is very close to

$$-A_1 - A_2 - A_3 = (-3\varepsilon, -1, -1, 1)^T.$$

Therefore, the column subset selection algorithm achieve about 4ε error on this matrix, which is a $4^{1-\frac{1}{p}}$ factor from being optimal. The similar construction works for any integer $k = 2^r - 1, r \in \mathbb{Z}^+$, where the lower bound is replaced by $(k+1)^{1-\frac{1}{p}}$, also matches with our upper bound exactly when $p \geq 2$.

2 Analysis of Approximation Ratio

In this section, we will prove Theorem 1.1. Recall that our goal is to bound $\mathrm{Err}(A_{J^*})$. We first introduce two useful lemmas. Lemma 2.1 gives an upper bound on approximation error by choosing a single arbitrary column subset A_J. Lemma 2.2 is our main technical lemma.

Lemma 2.1. *If J satisfies $\det(V_J) \neq 0$, the approximation error of A_J is upper bounded by*

$$\mathrm{Err}^p(A_J) \leq \|\Delta - \Delta_J V_J^{-1} V\|_p^p.$$

Lemma 2.2. *Let $S = \{s_{ij}\} \in \mathbb{C}^{k\times m}$ be a complex matrix, $r = (r_1, \cdots, r_m)$ be m-dimensional complex vector, and $T = \begin{bmatrix} r \\ S \end{bmatrix} \in \mathbb{C}^{(k+1)\times m}$, we have*

$$\sum_{I\in[m]^{k+1}} |\det(T_I)|^p \leq C_{p,k} \sum_{l=1}^{m} |r_l|^p \sum_{J\in[m]^k} |\det(S_J)|^p,$$

where

$$C_{p,k} = c_{p,k}^p = \begin{cases} (k+1), & 1 \leq p \leq 2, \\ (k+1)^{p-1}, & p \geq 2. \end{cases}$$

Now let us move to the proof of Theorem 1.1.

Proof. of Theorem 1.1: We can WLOG assume that $\mathrm{rank}(V) = k$. In fact, if $\mathrm{rank}(A) < k$, then of course $\mathrm{Err}^p(A_{J^*}) = 0$ and there is nothing to prove. Otherwise if $\mathrm{rank}(A) \geq k$, then by the definition of V, we know that $\mathrm{rank}(V) = k$.

We will upper bound the approximation error of the best column by a weighted average of $\mathrm{Err}^p(A_J)$. In other words, we are going to choose a set of non-negative weights w_J such that $\sum_{J\in[m]^k} w_J = 1$, and upper bound $\mathrm{Err}^p(A_{J^*})$ by

$$\mathrm{Err}^p(A_{J^*}) = \min_{J\in[m]^k} \mathrm{Err}^p(A_J) \leq \sum_{J\in[m]^k} w_J \mathrm{Err}^p(A_J).$$

In the following analysis, our choice of w_J will be

$$w_J = \frac{|\det(V_J)|^p}{\sum_{I\in[m]^k} |\det(V_I)|^p}.$$

Since rank$(V) = k$, w_J are well-defined. We first prove

$$|\det(V_J)|^p \mathrm{Err}^p(A_{J^*}) \le \sum_{d=1}^{n} \sum_{i=1}^{m} \left| \det \begin{pmatrix} \Delta_{dL} \\ V_L \end{pmatrix} \right|^p , \tag{3}$$

where we denote $L = (i, J) = (i, j_1, \cdots, j_k) \in [m]^{k+1}$.

In fact, when det$(V_J) = 0$, of course LHS of (3) = 0 $\le$ RHS. When det$(V_J) \neq 0$, we know that V_J is invertible. By Lemma 2.1,

$$\begin{aligned} |\det(V_J)|^p \mathrm{Err}^p(A_{J^*}) \le & |\det(V_J)|^p \|\Delta - \Delta_J V_J^{-1} V\|_p^p \\ =& \| \det(V_J) \left(\Delta - \Delta_J V_J^{-1} V\right) \|_p^p \\ =& \sum_{i=1}^{m} \| \det(V_J) \left(\Delta_i - \Delta_J V_J^{-1} V_i\right) \|_p^p \\ =& \sum_{d=1}^{n} \sum_{i=1}^{m} | \det(V_J) \left(\Delta_{di} - \Delta_{dJ} V_J^{-1} V_i\right) |^p \\ =& \sum_{d=1}^{n} \sum_{i=1}^{m} \left| \det \begin{pmatrix} \Delta_{di} & \Delta_{dJ} \\ V_i & V_J \end{pmatrix} \right|^p \\ =& \sum_{d=1}^{n} \sum_{i=1}^{m} \left| \det \begin{pmatrix} \Delta_{dL} \\ V_L \end{pmatrix} \right|^p . \end{aligned}$$

The second last equality follows from the Schur's determinant identity. Therefore (3) holds, and

$$\begin{aligned} \mathrm{Err}^p(A_{J^*}) \le & \sum_{J \in [m]^k} \frac{|\det(V_J)|^p}{\sum_{I \in [m]^k} |\det(V_I)|^p} \mathrm{Err}^p(A_J) \\ \le & \sum_{d=1}^{n} \left(\frac{1}{\sum_{I \in [m]^k} |\det(V_I)|^p} \sum_{J \in [m]^k} \sum_{i=1}^{m} \left| \det \begin{pmatrix} \Delta_{dL} \\ V_L \end{pmatrix} \right|^p \right) \\ = & \sum_{d=1}^{n} \left(\frac{1}{\sum_{I \in [m]^k} |\det(V_I)|^p} \sum_{L \in [m]^{k+1}} \left| \det \begin{pmatrix} \Delta_{dL} \\ V_L \end{pmatrix} \right|^p \right) . \end{aligned}$$

By Lemma 2.2,

$$\frac{1}{\sum_{I \in [m]^k} |\det(V_I)|^p} \sum_{L \in [m]^{k+1}} \left| \det \begin{pmatrix} \Delta_{dL} \\ V_L \end{pmatrix} \right|^p \le C_{p,k} \sum_{j=1}^{m} |\Delta_{dj}|^p .$$

Therefore,

$$\mathrm{Err}^p(A_{J^*}) \le \sum_{d=1}^{n} \left(C_{p,k} \sum_{j=1}^{m} |\Delta_{dj}|^p \right) = C_{p,k} \sum_{j=1}^{m} \|\Delta_j\|_p^p = C_{p,k} \mathrm{OPT}^p ,$$

which means

$$\mathrm{Err}(A_{J^*}) \le C_{p,k}^{1/p} \mathrm{OPT} = c_{p,k} \mathrm{OPT}.$$

□

Therefore, we only need to prove the two lemmas.

Proof. of Lemma 2.1: Recall that by definition of Δ_i, $A_i = UV_i + \Delta_i$,

$$\begin{aligned} \mathrm{Err}^p(A_J) &= \min_{Y \in \mathbb{R}^{k \times m}} \|A - A_J Y\|_p^p \\ &\le \|A - A_J V_J^{-1} V\|_p^p \\ &= \|(UV + \Delta) - (UV_J + \Delta_J) V_J^{-1} V\|_p^p \\ &= \|\Delta - \Delta_J V_J^{-1} V\|_p^p . \end{aligned}$$

$\square$

Proof. of Lemma 2.2: The proof is based on Riesz-Thorin interpolation theorem from harmonic analysis. We state a simplified version of the theorem below, which is the most convenient-to-use version for our proof. The general version can be found in the Appendix.

Lemma 2.3. *[Simplified version of Riesz-Thorin] Let* $\Lambda : \mathbb{C}^{n_1} \times \mathbb{C}^{n_2} \to \mathbb{C}^{n_0}$ *be a multi-linear operator, such that*

$$\|\Lambda(a,b)\|_{p_0} \leq M_{p_0} \|a\|_{p_0} \|b\|_{p_0}.$$
$$\|\Lambda(a,b)\|_{p_1} \leq M_{p_1} \|a\|_{p_1} \|b\|_{p_1}.$$

for all $a \in \mathbb{C}^{n_1}, b \in \mathbb{C}^{n_2}$*, then*

$$\|\Lambda(a,b)\|_{p_\theta} \leq M_{p_0}^{1-\theta} M_{p_1}^{\theta} \|a\|_{p_\theta} \|b\|_{p_\theta}.$$

holds for all $\theta \in [0,1]$*, where*

$$\frac{1}{p_\theta} := \frac{1-\theta}{p_0} + \frac{\theta}{p_1}.$$

Riesz-Thorin theorem is a classical result in interpolation theory that gives bounds for L^p to L^q operator norm. In general, it is easier to prove estimates within spaces like L^2, L^1 and L^∞. Interpolation theory enables us to generalize results in those spaces to some L^p and L^q spaces in between with an explicit operator norm. In our application, the X_i is a set of n_i elements and V_i is $\mathbb{C}^{n_i}$, the space of functions on n_i elements.

Now we prove Lemma 2.2. In fact, by symmetricity, lemma 2.2 is equivalent to

$$(k+1)! \sum_{I \in \binom{[m]}{k+1}} |\det(T_I)|^p \leq k! C_{p,k} \sum_{t=1}^{m} |r_t|^p \sum_{J \in \binom{[m]}{k}} |\det(S_J)|^p .$$

Here, $\binom{[m]}{k} = \{(i_1, \cdots, i_k) | 1 \leq i_1 < i_2 < \cdots < i_k \leq m\}$.

Taking $\frac{1}{p}$-th power on both sides, we have the following equivalent form

$$\left(\sum_{I \in \binom{[m]}{k+1}} |\det(T_I)|^p \right)^{\frac{1}{p}} \leq \frac{c_{p,k}}{(k+1)^{\frac{1}{p}}} \left(\sum_{t=1}^{m} |r_t|^p \right)^{\frac{1}{p}} \left(\sum_{J \in \binom{[m]}{k}} |\det(S_J)|^p \right)^{\frac{1}{p}} .$$

By Laplace expansion on the first row of $\det(T_I)$, we have for every $I = (i_1, \cdots, i_{k+1}) \in \binom{[m]}{k+1}$

$$\det(T_I) = \sum_{t=1}^{k+1} (-1)^{t+1} r_{i_t} \det(S_{I_{-t}}).$$

Here, $I_{-t} = (i_1, \cdots, i_{t-1}, i_{t+1}, \cdots, i_{k+1}) \in \binom{[m]}{k}$.

This motivates us to define the following multilinear map $\Lambda : \mathbb{C}^{\binom{[m]}{1}} \times \mathbb{C}^{\binom{[m]}{k}} \to \mathbb{C}^{\binom{[m]}{k+1}}$: for all $\{a_t\}_{t \in \binom{[m]}{1}} \in \mathbb{C}^{\binom{[m]}{1}}, \{b_J\}_{J \in \binom{[m]}{k}} \in \mathbb{C}^{\binom{[m]}{k}}$, and index set $I = (i_1, \cdots, i_{k+1}) \in \binom{[m]}{k+1}$, $[\Lambda(a,b)]_I$ is defined as

$$[\Lambda(a,b)]_I = \sum_{t=1}^{k+1} (-1)^{t+1} a_{i_t} b_{I_{-t}}.$$

Now, by letting $a_t = r_t, b_J = \det(S_J)$, the inequality can be written as

$$\|\Lambda(a,b)\|_p \leq \frac{c_{p,k}}{(k+1)^{\frac{1}{p}}} \|a\|_p \|b\|_p = \max\left(1, (k+1)^{1-\frac{2}{p}}\right) \|a\|_p \|b\|_p. \tag{4}$$

Let $M_p = \max\left(1, (k+1)^{1-\frac{2}{p}}\right)$, this is exactly the same form as Riesz-Thorin Theorem! We will first prove the easier cases $p = 1$,$p = 2$ and $p = \infty$. For general values of p, we denote

$$\frac{1}{p} = \frac{1-\theta}{p_0} + \frac{\theta}{p_1}.$$

here, when $p \in [1, 2)$, we choose $p_0 = 1, p_1 = 2$; when $p \in [2, +\infty)$, we choose $p_0 = 2, p_1 = +\infty$. We can easily verify that $M_p = M_{p_0}^{1-\theta} M_{p_1}^{\theta}$.By using the above two lemmas, we *only* need to show (4) holds for $p = 1, 2, \infty$.

We leave the complete proof of (4) in the appendix. □

3 Lower Bounds

In this section, we give a proof sketch of Theorem 1.2. The proof is constructive: we prove the theorem by showing for all $\varepsilon > 0$, we can construct a matrix $A(\varepsilon)$, such that selecting every k columns of $A(\varepsilon)$ leads to an approximation ratio at least $\frac{(k+1)^{1-\frac{1}{p}}}{1+o_\varepsilon(1)}$. Then, the theorem follows by letting $\varepsilon \to 0^+$. Our choice of $A(\varepsilon)$ is a perturbation of Hadamard matrices.

Throughout the proof, we assume that $k = 2^r - 1$, for some $r \in \mathbb{Z}^+$, and $\varepsilon > 0$ is an arbitrarily small constant. We consider the well known Hadamard matrix of order $(k+1) = 2^r$, defined below:

$$H^{(1)} = 1,$$
$$H^{(2^l)} = \begin{pmatrix} H^{(2^{l-1})} & H^{(2^{l-1})} \\ H^{(2^{l-1})} & -H^{(2^{l-1})} \end{pmatrix}, l \geq 1.$$

Now we can define $A(\varepsilon)$, the construction of lower bound instance: it is a perturbation of H by replacing all the entries on the first row by ε, i.e.,

$$A(\varepsilon)_{ij} = \begin{array}{cc} \varepsilon & \text{when} \quad i = 1, \\ H_{ij} & \text{when} \quad i \neq 1. \end{array}$$

We can see that $A(\varepsilon)$ is close to a rank-k matrix. In fact, $A(0)$ has rank at most k. Therefore, we can upper bound OPT by

$$\text{OPT} \leq \|A(\varepsilon) - A(0)\|_p = ((k+1)\varepsilon^p)^{1/p} = (k+1)^{1/p}\varepsilon. \tag{5}$$

The remaining work is to give a lower bound on the approximation error using any k columns. For simplicity of notations, we use A as shorthand for $A(\varepsilon)$ when it's clear from context. Say we are using all $(k+1)$ columns except the j-th, i.e. the column subset is $A_{[k+1]-\{j\}}$. Obviously, we achieve zero error on all the columns other than the j-th. Therefore, the approximation error is essentially the ℓ_p distance from A_j to $\text{span}(A_{[k+1]-\{j\}})$. We can show that the ℓ_p projection from A_j to $\text{span}(A_{[k+1]-\{j\}})$ is very close to $\sum_{i \neq j}(-A_i)$, in other words,

$$\text{Err}(A_{[k+1]-\{j\}}) = (1 - o(1))\|A_j - \sum_{i \neq j}(-A_i)\|_p = (1 - o(1))(k+1)\varepsilon \tag{6}$$

The theorem follows by combining (5) and (6). The complete proof can be found in the appendix.

4 Analysis of Efficient Algorithms

One drawback of the column subset selection algorithm is its time complexity - it requires $O(m^k \,\mathsf{poly}(n))$ time, which is not desirable since it's exponential in k. However, several more efficient algorithms [13] are designed based on it. Our tighter analysis on Algorithm 1 implies better approximation guarantees on these algorithms as well. Our improved bounds can be stated as follows:

Theorem 4.1. *Algorithm 2, which runs in* $\mathsf{poly}(m, n, k)$ *time and selects* $k \log m$ *columns, is a bi-criteria* $O(c_{p,k}) = O((k+1)^{\max(1/p, 1-1/p)})$ *approximation algorithm.*

Theorem 4.2. *Algorithm 3, which runs in* $\mathsf{poly}(m, n)$ *time as long as* $k = O(\frac{\log n}{\log \log n})$ *is an* $O(c_{p,k}^3 k \log m) = O(k^{\max(1+3/p, 4-3/p)} \log m)$ *approximation algorithm.*

These results improve the previous $O(k)$ and $O(k^4 \log(m))$ bounds respectively. We include the proof and the description of Algorithm 2 and Algorithm 3 in Appendix for completeness.

References

[1] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[2] Peter J Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.

[3] Lei Xu and Alan L Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, 6(1):131–143, 1995.

[4] Deyu Meng and Fernando De La Torre. Robust matrix factorization with unknown noise. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1337–1344, 2013.

[5] Naiyan Wang and Dit-Yan Yeung. Bayesian robust matrix factorization for image and video processing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1785–1792, 2013.

[6] Liang Xiong, Xi Chen, and Jeff Schneider. Direct robust matrix factorizatoin for anomaly detection. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 844–853. IEEE, 2011.

[7] Nicolas Gillis and Stephen A Vavasis. On the complexity of robust pca and l1-norm low-rank matrix approximation. *Mathematics of Operations Research*, 2018.

[8] Zhao Song, David P Woodruff, and Peilin Zhong. Low rank approximation with entrywise l 1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 688–701. ACM, 2017.

[9] Nicolas Gillis and Yaroslav Shitov. Low-rank matrix approximation in the infinity norm. *arXiv preprint arXiv:1706.00078*, 2017.

[10] Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P Woodruff. A ptas for lp-low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 747–766. SIAM, 2019.

[11] Anastasios Kyrillidis. Simple and practical algorithms for lp norm low-rank approximation. *arXiv preprint arXiv:1805.09464*, 2018.

[12] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*, 2017.

[13] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P Woodruff. Algorithms for lp low rank approximation. *arXiv preprint arXiv:1705.06730*, 2017.

[14] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006.

[15] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

[16] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 292–303. Springer, 2006.

[17] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. SIAM, 2009.

[18] Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 329–338. IEEE, 2010.

[19] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[20] Christos Boutsidis and David P Woodruff. Optimal cur matrix decompositions. *SIAM Journal on Computing*, 46(2):543–589, 2017.

[21] Yining Wang and Aarti Singh. Column subset selection with missing data via active sampling. In *Artificial Intelligence and Statistics*, pages 1033–1041, 2015.

[22] Aditya Bhaskara and Silvio Lattanzi. Non-negative sparse regression and column subset selection with l1 error. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 94. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[23] Chen Dan, Kristoffer Arnsfelt Hansen, He Jiang, Liwei Wang, and Yuchen Zhou. Low Rank Approximation of Binary Matrices: Column Subset Selection and Generalizations. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-086-6. doi: 10.4230/LIPIcs.MFCS.2018.41. URL `http://drops.dagstuhl.de/opus/volltexte/2018/9623`.

[24] Qifa Ke and Takeo Kanade. Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.

[25] Feiping Nie, Jianjun Yuan, and Heng Huang. Optimal mean robust principal component analysis. In *International conference on machine learning*, pages 1062–1070, 2014.

[26] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

[27] Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit Dhillon. Robust principal component analysis with side information. In *International Conference on Machine Learning*, pages 2291–2299, 2016.

[28] Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust pca via gradient descent. In *Advances in neural information processing systems*, pages 4152–4160, 2016.

[29] Karl Bringmann, Pavel Kolev, and David Woodruff. Approximation algorithms for l0-low rank approximation. In *Advances in Neural Information Processing Systems*, pages 6648–6659, 2017.

[30] Ricardo Otazo, Emmanuel Candès, and Daniel K Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 73(3):1125–1136, 2015.

[31] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

[32] Javad Mashreghi. *Representation theorems in Hardy spaces*, volume 74. Cambridge University Press, 2009.

[33] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687–717, 2014.

[34] Yining Wang and Aarti Singh. Provably correct algorithms for matrix column subset selection with selectively sampled data. *arXiv preprint arXiv:1505.04343*, 2015.

[35] Zhao Song, David P Woodruff, and Peilin Zhong. Relative error tensor low rank approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2772–2789. SIAM, 2019.

[36] Zhao Song, David P Woodruff, and Peilin Zhong. Towards a zero-one law for entrywise low rank approximation. *arXiv preprint arXiv:1811.01442*, 2018.

[37] Amit Deshpande, Madhur Tulsiani, and Nisheeth K Vishnoi. Algorithms and hardness for subspace approximation. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 482–496. Society for Industrial and Applied Mathematics, 2011.

[38] Ravindran Kannan, Santosh Vempala, et al. Spectral algorithms. *Foundations and Trends® in Theoretical Computer Science*, 4(3–4):157–288, 2009.

A Riesz-Thorin Interpolation Theorem

Lemma A.1 (Riesz-Thorin interpolation theorem , see Lemma 8.5 in [32]). *Let $(X_i, \mathfrak{M}_i, \mu_i)$, $i = 0, 1, 2, \cdots, n$ be measure spaces. Let V_i represent the complex vector space of simple functions on X_i. Suppose that*

$$\Lambda : V_1 \times V_2 \times \cdots \times V_n \to V_0.$$

is a multi-linear operator of types p_0 and p_1 where $p_0, p_1 \in [1, \infty]$, with constants M_0 and M_1, respectively. i.e.,

$$\|\Lambda(f_1, f_2, \cdots, f_n)\|_{p_i} \leq M_{p_i} \|f_1\|_{p_i} \|f_2\|_{p_i} \cdots \|f_n\|_{p_i}.$$

for $i = 0, 1$. Let $\theta \in [0, 1]$ and define

$$\frac{1}{p_\theta} := \frac{1-\theta}{p_0} + \frac{\theta}{p_1}.$$

Then, Λ is of type p_θ with constant $M_{p_\theta} := M_{p_0}^{1-\theta} M_{p_1}^{\theta}$, that is,

$$\|\Lambda(f_1, f_2, \cdots, f_n)\|_{p_\theta} \leq M_{p_\theta} \|f_1\|_{p_\theta} \|f_2\|_{p_\theta} \cdots \|f_n\|_{p_\theta}.$$

Lemma 2.3 is a direct corollary of this theorem.

B Lower Bounds

In this section, we will prove Theorem 1.2. The proof is constructive: we prove the theorem by showing for all $\varepsilon > 0$, we can construct a matrix $A(\varepsilon)$, such that selecting every k columns of $A(\varepsilon)$ leads to an approximation ratio at least $\frac{(k+1)^{1-\frac{1}{p}}}{(1+k\varepsilon^q)^{1/q}}$. Then, the theorem follows by letting $\varepsilon \to 0^+$. Our choice of $A(\varepsilon)$ is a perturbation of Hadamard matrices, defined below.

Throughout the proof of Theorem 1.2, we assume that $k = 2^r - 1$, for some $r \in \mathbb{Z}^+$, and $\varepsilon > 0$ is an arbitrarily small constant.

Proof. of Theorem 1.2: We consider the well known Hadamard matrix of order $(k+1) = 2^r$, defined below:

$$H^{(1)} = 1,$$

$$H^{(2^l)} = \begin{pmatrix} H^{(2^{l-1})} & H^{(2^{l-1})} \\ H^{(2^{l-1})} & -H^{(2^{l-1})} \end{pmatrix}, l \geq 1.$$

The Hadamard matrix has the following properties: (we will use H to represent $H^{(2^r)}$ when it's clear from context)

- $H_{di} = 1$ or $H_{di} = -1$.
- All entries on the first row are ones, i.e. $H_{1j} = 1$.
- The columns of H are pairwisely orthogonal, i.e.

  $$\sum_{d=1}^{k+1} H_{di} H_{dj} = 0$$

  holds when $i \neq j$.

Now we can define $A(\varepsilon)$: it is a perturbation of H by replacing all the entries on the first row by ε, i.e.,

$$A(\varepsilon)_{ij} = \begin{matrix} \varepsilon & \text{when} & i = 1, \\ H_{ij} & \text{when} & i \neq 1. \end{matrix}$$

We can see that $A(\varepsilon)$ is close to a rank-k matrix. In fact, $A(0)$ has rank at most k. Also, $A(0)$ is an $2^r \times 2^r$, or equivalently, $(k+1) \times (k+1)$ matrix, and it has all zeros on the first row. Therefore, we can upper bound OPT by

$$\mathrm{OPT} \le \|A(\varepsilon) - A(0)\|_p = ((k+1)\varepsilon^p)^{1/p} = (k+1)^{1/p}\varepsilon.$$

The remaining work is to give a lower bound on the approximation error using any k columns. For simplicity of notations, we use A as shorthand for $A(\varepsilon)$ when it's clear from context. Say we are using all $(k+1)$ columns except the j-th, i.e. the column subset is $A_{[k+1]-\{j\}}$. Obviously, we achieve zero error on all the columns other than the j-th. Therefore, the approximation error is essentially the ℓ_p distance from A_j to $\mathrm{span}(A_{[k+1]-\{j\}})$. Thus,

$$\begin{aligned}
\mathrm{Err}(A_{[k+1]-\{j\}}) &= \inf_{x_i \in \mathbb{R}} \|A_j - \sum_{i \neq j} x_i A_i\|_p \\
&= \inf_{x_i \in \mathbb{R}} \left(\sum_{d=1}^{k+1} \left| A_{dj} - \sum_{i \neq j} x_i A_{di} \right|^p \right)^{1/p} \\
&= \inf_{x_i \in \mathbb{R}} \left(\varepsilon^p \left| 1 - \sum_{i \neq j} x_i \right|^p + \sum_{d=2}^{k+1} \left| H_{dj} - \sum_{i \neq j} x_i H_{di} \right|^p \right)^{1/p}.
\end{aligned}$$

By Hölder's inequality,

$$\begin{aligned}
&\left(\varepsilon^p \left| 1 - \sum_{i \neq j} x_i \right|^p + \sum_{d=2}^{k+1} \left| H_{dj} - \sum_{i \neq j} x_i H_{di} \right|^p \right)^{1/p} \left(\left(\frac{1}{\varepsilon}\right)^q + \sum_{d=2}^{k+1} |H_{dj}|^q \right)^{1/q} \\
\ge &\left(1 - \sum_{i \neq j} x_i \right) + \sum_{d=2}^{k+1} H_{dj} \left(H_{dj} - \sum_{i \neq j} x_i H_{di} \right)
\end{aligned}$$

where $\frac{1}{p} + \frac{1}{q} = 1$.

We can actually show that $RHS = k+1$.

Using the fact that $H_{1i} = H_{1j} = 1$ and $\sum_{d=1}^{k+1} H_{di} H_{dj} = 0$,

$$\begin{aligned}
RHS &= \left(1 - \sum_{i \neq j} x_i \right) + \sum_{d=2}^{k+1} H_{dj} \left(H_{dj} - \sum_{i \neq j} x_i H_{di} \right) \\
&= \left(1 - \sum_{i \neq j} H_{1i} H_{1j} x_i \right) + \sum_{d=2}^{k+1} \left(1 - \sum_{i \neq j} x_i H_{di} H_{dj} \right) \\
&= \sum_{d=1}^{k+1} \left(1 - \sum_{i \neq j} x_i H_{di} H_{dj} \right) \\
&= (k+1) - \sum_{i \neq j} x_i \left(\sum_{d=1}^{k+1} H_{di} H_{dj} \right) \\
&= k+1.
\end{aligned}$$

Now we can finally bound the approximation error

$$\begin{aligned}\mathrm{Err}(A_{[k+1]-\{j\}}) &= \inf_{x_i\in\mathbb{R}}\left(\varepsilon^p\left|1-\sum_{i\neq j}x_i\right|^p+\sum_{d=2}^{k+1}\left|H_{dj}-\sum_{i\neq j}x_iH_{di}\right|^p\right)^{1/p}\\ &\geq \frac{k+1}{\left(\left(\frac{1}{\varepsilon}\right)^q+\sum_{d=2}^{k+1}|H_{dj}|^q\right)^{1/q}}\\ &= \frac{k+1}{\left(\varepsilon^{-q}+k\right)^{1/q}}\\ &= \frac{(k+1)\varepsilon}{\left(1+k\varepsilon^q\right)^{1/q}}.\end{aligned}$$

Thus,

$$\frac{\mathrm{Err}(A_{[k+1]-\{j\}})}{\mathrm{OPT}}\geq\frac{(k+1)^{1-\frac{1}{p}}}{\left(1+k\varepsilon^q\right)^{1/q}}.$$

Note that this bound can be arbitrarily close to $(k+1)^{1-\frac{1}{p}}$ when ε is small enough, thus we complete the proof. □

C Proof of Equation (4)

Now we are going to prove (4). First, we need to extend the definition of b_J for all $J=(j_1,\cdots,j_k)\in[m]^k$. This definition is similar to the property of determinants.

- When $1\leq j_1<j_2<\cdots<j_k\leq m$, i.e. $J\in\binom{[m]}{k}$, b_J is already defined.
- When there exists $s\neq t, j_s=j_t$, define $b_J=0$.
- Otherwise, there exists $1\leq j'_1<j'_2<\cdots<j'_k\leq m$ and a permutation π, such that

$$(j_1,\cdots,j_k)=\pi(j'_1,j'_2,\cdots,j'_k).$$

  Let $J'=(j'_1,j'_2,\cdots,j'_k)$. In such case, we define

$$b_J=\mathrm{sgn}(\pi)b_{J'},$$

  where $\mathrm{sgn}(\pi)$ is the parity of π, i.e. $\mathrm{sgn}(\pi)=1$ if π is an even permutation, and $\mathrm{sgn}(\pi)=-1$ otherwise.

Note that if J is a transposition (2-element exchanges) of $\tilde{J}$, then $b_J=-b_{\tilde{J}}$.

We can also define $[\Lambda(a,b)]_I$ for all $I\in[m]^{k+1}$, by

$$[\Lambda(a,b)]_I=\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}.$$

Here, $I_{-t}=(i_1,\cdots,i_{t-1},i_{t+1},\cdots,i_{k+1})\in[m]^k$. Similarly, if I is a transposition (2-element exchanges) of $\tilde{I}$, then $[\Lambda(a,b)]_I=-[\Lambda(a,b)]_{\tilde{I}}$.

As mentioned before, we only need to verify (4) for the special cases $p=1,2,\infty$.

Case 1: $p=1$. The inequality is equivalent to

$$\|\Lambda(a,b)\|_1\leq\|a\|_1\|b\|_1.$$

In fact, by the definition, we always have

$$\|\Lambda(a,b)\|_1=\sum_{I\in\binom{[m]}{k+1}}|[\Lambda(a,b)]_I|=\frac{1}{(k+1)!}\sum_{I\in[m]^{k+1}}|[\Lambda(a,b)]_I|.$$

Therefore,

$$\begin{aligned}
\|\Lambda(a,b)\|_1 &= \frac{1}{(k+1)!}\sum_{I\in[m]^{k+1}}\left|\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}\right| \\
&\le \frac{1}{(k+1)!}\sum_{I\in[m]^{k+1}}\sum_{t=1}^{k+1}|a_{i_t}||b_{I_{-t}}| \\
&= \frac{1}{(k+1)!}(k+1)\sum_{I\in[m]^{k+1}}|a_{i_1}||b_{I_{-1}}| \\
&= \frac{1}{k!}\sum_{i_1\in[m]}|a_{i_1}|\sum_{J\in[m]^k}|b_J| \\
&= \sum_{i_1\in[m]}|a_{i_1}|\sum_{J\in\binom{[m]}{k}}|b_J| \\
&= \|a\|_1\|b\|_1.
\end{aligned}$$

Case 2: $p=\infty$. The inequality is equivalent to

$$\|\Lambda(a,b)\|_\infty \le (k+1)\|a\|_\infty\|b\|_\infty$$

$$\begin{aligned}
\|\Lambda(a,b)\|_\infty &= \max_{I\in\binom{[m]}{k+1}}|[\Lambda(a,b)]_I| \\
&= \max_{I\in\binom{[m]}{k+1}}\left|\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}\right| \\
&\le \max_{I\in\binom{[m]}{k+1}}\sum_{t=1}^{k+1}|a_{i_t}||b_{I_{-t}}| \\
&\le \sum_{t=1}^{k+1}\max_{i_t\in[m]}|a_{i_t}|\max_{J\in\binom{[m]}{k}}|b_J| \\
&= (k+1)\max_{i_1\in[m]}|a_{i_1}|\max_{J\in\binom{[m]}{k}}|b_J| \\
&= (k+1)\|a\|_\infty\|b\|_\infty.
\end{aligned}$$

Case 3: $p=2$. The inequality is equivalent to

$$\|\Lambda(a,b)\|_2 \le \|a\|_2\|b\|_2.$$

$$\begin{aligned}
\|\Lambda(a,b)\|_2^2 &= \sum_{I\in\binom{[m]}{k+1}}|[\Lambda(a,b)]_I|^2 \\
&= \frac{1}{(k+1)!}\sum_{I\in[m]^{k+1}}|[\Lambda(a,b)]_I|^2 \\
&= \frac{1}{(k+1)!}\sum_{I\in[m]^{k+1}}\left|\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}\right|^2.
\end{aligned}$$

Note that

$$\begin{aligned}\left|\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}\right|^2 &= \left(\sum_{t=1}^{k+1}(-1)^{t+1}a_{i_t}b_{I_{-t}}\right)\left(\sum_{s=1}^{k+1}(-1)^{s+1}\bar{a}_{i_s}\bar{b}_{I_{-s}}\right)\\ &= \sum_{t=1}^{k+1}|a_{i_t}|^2|b_{I_{-t}}|^2 + 2\sum_{1\le t\neq s\le k+1}(-1)^{t+s}a_{i_t}b_{I_{-t}}\bar{a}_{i_s}\bar{b}_{I_{-s}}.\end{aligned}$$

Therefore,

$$(k+1)!\|\Lambda(a,b)\|_2^2 = \sum_{I\in[m]^{k+1}}\sum_{t=1}^{k+1}|a_{i_t}|^2|b_{I_{-t}}|^2 + 2\sum_{I\in[m]^{k+1}}\sum_{1\le t\neq s\le k+1}(-1)^{t+s}a_{i_t}b_{I_{-t}}\bar{a}_{i_s}\bar{b}_{I_{-s}}.$$

The first term can be simplified as

$$\begin{aligned}&\sum_{I\in[m]^{k+1}}\sum_{t=1}^{k+1}|a_{i_t}|^2|b_{I_{-t}}|^2\\ &= (k+1)\sum_{i_1\in[m]}|a_{i_1}|^2\sum_{J\in[m]^k}|b_J|^2\\ &= (k+1)!\|a\|_2^2\|b\|_2^2.\end{aligned}$$

Therefore, we only need to prove that the second term is non-positive.

When $t < s$,

$$\begin{aligned}b_{I_{-s}} &= b_{(i_1,\cdots,i_{s-1},i_{s+1},\cdots,i_{k+1})}\\ &= (-1)^{t-1}b_{(i_l,i_1,\cdots,i_{t-1},i_{t+1},\cdots,i_{s-1},i_{s+1},\cdots,i_{k+1})}\\ &= (-1)^{t-1}b_{(i_t,I_{-\{t,s\}})},\end{aligned}$$

and

$$\begin{aligned}b_{I_{-t}} &= b_{(i_1,\cdots,i_{t-1},i_{t+1},\cdots,i_{k+1})}\\ &= (-1)^{s-2}b_{(i_s,i_1,\cdots,i_{t-1},i_{t+1},\cdots,i_{s-1},i_{s+1},\cdots,i_{k+1})}\\ &= (-1)^{s-2}b_{(i_s,I_{-\{t,s\}})}.\end{aligned}$$

Therefore,

$$(-1)^{t+s}b_{I_{-t}}\bar{b}_{I_{-s}} = -b_{(i_s,I_{-\{t,s\}})}\bar{b}_{(i_t,I_{-\{t,s\}})}.$$

The same argument holds for the case $t > s$. Thus, for each pair of (t,s), we have

$$\begin{aligned}&\sum_{I\in[m]^{k+1}}(-1)^{t+s}a_{i_t}b_{I_{-t}}\bar{a}_{i_s}\bar{b}_{I_{-s}}\\ &= -\sum_{I\in[m]^{k+1}}a_{i_t}\bar{a}_{i_s}b_{(i_s,I_{-\{t,s\}})}\bar{b}_{(i_t,I_{-\{t,s\}})}\\ &= -\sum_{J\in[m]^{k-1}}\sum_{i_t=1}^{m}\sum_{i_s=1}^{m}a_{i_t}\bar{a}_{i_s}b_{(i_s,J)}\bar{b}_{(i_t,J)}\\ &= -\sum_{J\in[m]^{k-1}}\left(\sum_{i_t=1}^{m}a_{i_t}\bar{b}_{(i_t,J)}\right)\left(\sum_{i_s=1}^{m}\bar{a}_{i_s}b_{(i_s,J)}\right)\\ &= -\sum_{J\in[m]^{k-1}}\left|\sum_{i_t=1}^{m}a_{i_t}\bar{b}_{(i_t,J)}\right|^2.\end{aligned}$$

Thus, the second term can be simplified as

Algorithm 2 SELECTCOLUMNS (k, A): Selecting $O(k \log m)$ columns of A.

```
Input: Data matrix $A$ and rank parameter $k$.
Output: $O(k \log m)$ columns of $A$
if number of columns of $A \leq 2k$ then
    return all the columns of $A$
else
    repeat
        Let $R$ be uniform at random $2k$ columns of $A$
    until at least $(1/10)$-fraction columns of $A$ are $\lambda_p$-approximately covered
    Let $A_{\overline{R}}$ be the columns of $A$ not approximately covered by $R$
    return $A_R \cup$ SELECTCOLUMNS $(k, A_{\overline{R}})$
end if
```

$$\begin{aligned}
&\sum_{I \in [m]^{k+1}} \sum_{1 \leq t \neq s \leq k+1} (-1)^{t+s} 2 a_{i_t} b_{I_{-t}} \bar{a}_{i_s} \bar{b}_{I_{-s}} \\
&= 2 \sum_{1 \leq t \neq s \leq k+1} \sum_{I \in [m]^{k+1}} (-1)^{t+s} a_{i_t} b_{I_{-t}} \bar{a}_{i_s} \bar{b}_{I_{-s}} \\
&= -4k(k+1) \sum_{J \in [m]^{k-1}} \left| \sum_{i_l=1}^{m} a_{i_l} \bar{b}_{(i_l, J)} \right|^2 \leq 0.
\end{aligned}$$

□

D Analysis for A poly(nm)-Time Bi-Criteria Algorithm

We can prove that Algorithm 2 from [13] runs in time poly(nm) but returns $O(k \log m)$ columns of A that can be used in place of U, with an error $O(c_{p,k})$ times the error of the best k-factorization. In other words, it obtains more than k columns but achieves a polynomial running time. The analysis can derived by slightly modifying the definition and proof in [13].

Definition 1 (Approximate coverage)**.** *Let S be a subset of k column indices. We say that column A_i is λ_p-**approximately covered** by S if for $p \in [1, \infty)$ we have $\min_{x \in \mathbb{R}^{k \times 1}} \|A_S x - A_i\|_p^p \leq \lambda \frac{100 c_{p,k}^p \|\Delta\|_p^p}{n}$, and for $p = \infty$, $\min_{x \in \mathbb{R}^{k \times 1}} \|A_S x - A_i\|_\infty \leq \lambda (k+1) \|\Delta\|_\infty$. If $\lambda = 1$, we say A_i is **covered** by S.*

We first show that if we select a set R columns of size $2k$ uniformly at random in $\binom{[m]}{2k}$, with constant probability we cover a constant fraction of columns of A.

Lemma D.1. *Suppose R is a set of $2k$ uniformly random chosen columns of A. With probability at least $2/9$, R covers at least a $1/10$ fraction of columns of A.*

Proof. Same as the proof of Lemma 6 in [13] except that we use $c_{p,k}^p$ instead of $(k+1)$ in the approximation bounds.

□

We are now ready to introduce Algorithm 2. As mentioned in [13], we can without loss of generality assume that the algorithm knows a number N for which $|\Delta|_p \leq N \leq 2|\Delta|_p$. Indeed, such a value can be obtained by first computing $|\Delta|_2$ using the SVD. Note that although one does not know Δ, one does know $|\Delta|_2$ since this is the Euclidean norm of all but the top k singular values of A, which one can compute from the SVD of A. Then, note that for $p < 2$, $|\Delta|_2 \leq |\Delta|_p \leq n^{2-p}|\Delta|_2$, while for $p \geq 2$, $|\Delta|_p \leq |\Delta|_2 \leq n^{1-2/p}|\Delta|_p$. Hence, there are only $O(\log n)$ values of N to try, given $|\Delta|_2$, one of which will satisfy $|\Delta|_p \leq N \leq 2|\Delta|_p$. One can take the best solution found by Algorithm 2 for each of the $O(\log n)$ guesses to N.

Theorem D.2. *With probability at least $9/10$, Algorithm 2 runs in time* poly(nm) *and returns $O(k \log m)$ columns that can be used as a factor of the whole matrix inducing ℓ_p error $O(c_{p,k}|\Delta|_p)$.*

Algorithm 3 An algorithm that transforms an $O(k \log m)$-rank matrix factorization into a k-rank matrix factorization without inflating the error too much.

```
1: Input: $U \in \mathbb{R}^{n \times O(k \log m)}$, $V \in \mathbb{R}^{O(k \log m) \times m}$
2: Output: $W \in \mathbb{R}^{n \times k}$, $Z \in \mathbb{R}^{k \times m}$
3: Apply Lemma E.1 to $U$ to obtain matrix $W^0$
4: Run $\ell_p$ linear regression over $Z^0$, s.t. $\|W^0 Z^0 - UV\|_p$ is minimized
5: Apply Algorithm 1 with input $(Z^0)^T \in \mathbb{R}^{n \times O(k \log m)}$ and $k$ to obtain $X$ and $Y$
6: Set $Z \leftarrow X^T$
7: Set $W \leftarrow W^0 Y^T$
8: Output $W$ and $Z$
```

Proof. Same as the proof of Theorem 7 in [13] except that we use $c_{p,k}^p$ instead of $(k+1)$ in the approximation bounds. □

E Analysis for A $((k \log n)^k \operatorname{poly}(mn))$-Time Algorithm

In this section we show how to get a rank-k, $O(c_{p,k}^3 k \log m)$-approximation efficiently starting from a rank-$O(k \log m)$ approximation. This algorithm runs in polynomial time as long as $k = O\left(\frac{\log n}{\log \log n}\right)$.

Let U be the columns of A selected by Algorithm 2.

E.1 An Isoperimetric Transformation

The first step of the proof is to show that we can modify the selected columns of A to span the same space but to have small distortion. For this, we need the following notion of isoperimetry.

Definition 2 (Almost isoperimetry). *A matrix $B \in \mathbb{R}^{n \times m}$ is* almost-ℓ_p-isoperimetric *if for all x, we have*

$$\frac{\|x\|_p}{2m} \leq \|Bx\|_p \leq \|x\|_p.$$

The following lemma from [13] show that given a full rank $A \in \mathbb{R}^{n \times m}$, it is possible to construct in polynomial time a matrix $B \in \mathbb{R}^{n \times m}$ such that A and B span the same space and B is almost-ℓ_p-isoperimetric.

Lemma E.1 (Lemma 10 in [13]). *Given a full (column) rank $A \in \mathbb{R}^{n \times m}$, there is an algorithm that transforms A into a matrix B such that $\operatorname{span} A = \operatorname{span} B$ and B is almost-ℓ_p-isoperimetric. Furthermore the running time of the algorithm is poly(nm).*

E.2 Reducing the Rank to k

Here we give an analysis of Algorithm 3 from [13]. It reduces the rank of our low-rank approximation from $O(k \log m)$ to k. Let $\delta = \|\Delta\|_p = \mathsf{OPT}$.

Theorem E.2. *Let $A \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times O(k \log m)}$, $V \in \mathbb{R}^{O(k \log m) \times m}$ be such that $\|A - UV\|_p = O(k\delta)$. Then, Algorithm 3 runs in time $O(k \log m)^k (mn)^{O(1)}$ and outputs $W \in \mathbb{R}^{n \times k}$, $Z \in \mathbb{R}^{k \times m}$ such that $\|A - WZ\|_p = O((c_{p,k}^3 k \log m)\delta)$.*

Proof. We start by bounding the running time. Step 3 is computationally the most expensive since it requires to execute a brute-force search on the $O(k \log m)$ columns of $(Z^0)^T$. So the running time is $O((k \log m)^k (mn)^{O(1)})$.

Now we have to show that the algorithm returns a good approximation. The main idea behind the proof is that UV is a low-rank approximable matrix. So after applying Lemma E.1 to U to obtain a low-rank approximation for UV we can simply focus on $Z^0 \in \mathbb{R}^{O(k \log m) \times n}$. Next, by applying Algorithm 1 to Z^0, we obtain a low-rank approximation in time $O(k \log m)^k (mn)^{O(1)}$. Finally we can use this solution to construct the solution to our initial problem.

We know by assumption that $\|A-UV\|_p = O(c_{p,k}\delta)$. Therefore, it suffices by the triangle inequality to show $\|UV - WZ\|_p = O(c_{p,k}^3 k \log m\delta)$. First note that $UV = W^0 Z^0$ since Lemma E.1 guarantees that $\mathrm{span}U = \mathrm{span}W^0$. Hence we can focus on proving $\|W^0 Z^0 - WZ\|_p \leq O((c_{p,k}^3 k \log m)\delta)$.

We first prove two useful intermediate steps.

Lemma E.3. *There exist matrices* $U^* \in \mathbb{R}^{n\times k}$, $V^* \in \mathbb{R}^{k\times m}$ *such that* $\|W^0 Z^0 - U^* V^*\|_p = O(c_{p,k}\delta)$.

Proof. Same as the proof of Lemma 12 in [13] except that we use $O(c_{p,k}\delta)$ instead of $O(k\delta)$. □

Lemma E.4. *There exist matrices* $F \in \mathbb{R}^{O(k\log m)\times k}$, $D \in \mathbb{R}^{k\times n}$ *such that* $\|W^0(Z^0 - FD)\|_p = O(c_{p,k}^2\delta)$.

Proof. Same as the proof of Lemma 13 in [13] except that we use $O(c_{p,k}\delta)$ and $O(c_{p,k}^2\delta)$instead of $O(k\delta)$ and $O(k^2\delta)$. □

Now from the guarantees of Lemma E.1 we know that for any vector y, $\|W^0 y\|_p \leq \frac{\|y\|_p}{k\log m}$. So we have $\|Z^0 - FD\|_p \leq O((c_{p,k}^2 k\log m)\delta)$, Thus $\|(Z^0)^T - D^T F^T\|_p \leq O((c_{p,k}^2 k\log m)\delta)$, so $(Z^0)^T$ has a low-rank approximation with error at most $O((c_{p,k}^2 k\log m)\delta)$. So we can apply Theorem 1.1 again and we know that there are k columns of $(Z^0)^T$ such that the low-rank approximation obtained starting from those columns has error at most $O((c_{p,k}^3 k\log m)\delta)$. We obtain such a low-rank approximation from Algorithm 1 with input $(Z^0)^T \in \mathbb{R}^{n\times O(k\log m)}$ and k. More precisely, we obtain an $X \in \mathbb{R}^{n\times k}$ and $Y \in \mathbb{R}^{k\times O(k\log m)}$ such that $\|(Z^0)^T - XY\|_p \leq O((c_{p,k}^3 k\log m)\delta)$. Thus $\|Z^0 - Y^T X^T\|_p \leq O((c_{p,k}^3 k\log m)\delta)$.

Now using again the guarantees of Lemma E.1 for W^0, we get $\|W^0(Z^0 - Y^T X^T)\|_p \leq O((c_{p,k}^3 k\log m)\delta)$. So $\|W^0(Z^0 - Y^T X^T)\|_p = \|W^0 Z^0 - WZ)\|_p = \|UV - WZ\|_p \leq O((c_{p,k}^3 k\log m)\delta)$. By combining it with $\|A - UV\|_p = O(c_{p,k}\delta)$ and using the Minkowski inequality, the proof is complete. □