

---

# Additive Error Guarantees for Weighted Low Rank Approximation

---

Aditya Bhaskara<sup>\*1</sup> Aravinda Kanchana Ruwanpathirana<sup>\*1</sup> Maheshakya Wijewardena<sup>\*1</sup>

## Abstract

Low-rank approximation is a classic tool in data analysis, where the goal is to approximate a matrix  $A$  with a low-rank matrix  $L$  so as to minimize the error  $\|A - L\|_F^2$ . However in many applications, approximating some entries is more important than others, which leads to the weighted low rank approximation problem. However, the addition of weights makes the low-rank approximation problem intractable. Thus many works have obtained efficient algorithms under additional structural assumptions on the weight matrix (such as low rank, and appropriate block structure). We study a natural greedy algorithm for weighted low rank approximation and develop a simple condition under which it yields bi-criteria approximation up to a small additive factor in the error. The algorithm involves iteratively computing the top singular vector of an appropriately varying matrix, and is thus easy to implement at scale. Our methods also allow us to study the problem of low rank approximation under  $\ell_p$  norm error.

## 1. Introduction

Matrix low rank approximation is one of the most classic dimension reduction methods in data analysis. The standard least squared error version can also be solved efficiently using the singular value decomposition, and we know how to do this in time comparable to the input sparsity (Clarkson & Woodruff, 2017). Despite its utility, natural variants of low-rank approximation turn out to be intractable. Weighted low-rank approximation is one well studied example: in many applications, some of the entries of a matrix may be less important to approximate than others (e.g., they might be known to be noisy), and thus we may have a weight associated with each entry. While standard least-squares regression for vectors can incorporate weights directly, the

matrix version turns out to be challenging. Formally, the weighted low-rank approximation problem is defined as follows: given  $A \in \mathbb{R}^{d \times n}$ , a non-negative weight matrix  $W \in \mathbb{R}^{d \times n}$  and a parameter  $k$ , the goal is to find a rank  $k$  matrix  $L$  that minimizes  $Cost(L)$ ,

$$Cost(L) := \sum_{i,j} W_{ij} \cdot (A_{ij} - L_{ij})^2. \quad (1)$$

The problem and its difficulty were recognized as early as Young (1941), and it has been well-studied in the ML literature starting with the work of Srebro & Jaakkola (2003). Unlike the unweighted version (which corresponds to  $W = \mathbf{1}_{d \times n}$ ) low-rank approximation, the problem above is NP-hard in general (Gillis & Glineur, 2011). Much of the early work such as Srebro & Jaakkola (2003); Manton et al. (2003); Eriksson & van den Hengel (2010) thus developed heuristics for the problem. The first provably efficient algorithms were obtained in the work of Razenshteyn et al. (2016) (see also references therein for work on matrix completion which is closely related). Razenshteyn et al. (2016) as well as more recent works (Musco et al., 2020; Ban et al., 2019b) assume that  $W$  has low rank, and develop algorithms that achieve a  $(1 + \epsilon)$  (multiplicative) approximation to the optimum cost, while having a running time exponential in the rank of  $W$ .

Very recently, Musco et al. (2020) initiated a study of additive error bounds for weighted low rank approximation. Here the goal is to obtain an  $L'$  such that  $Cost(L') \leq OPT + \epsilon \|A\|_F^2$ , where  $OPT$  is the optimal cost. Additive error guarantees have been a classic notion in the literature on low rank approximation (starting with the seminal work of Frieze et al. (2004) on sampling for low rank approximation with additive error). Additive guarantees are realistic in applications where the optimal error is a small yet constant fraction of the total mass (e.g., when a low rank approximation may capture 90% of the Frobenius mass).

So far, our discussion has been restricted to error in the squared norm. However, low rank approximation has also been studied in entrywise  $\ell_p$  norms for  $p \neq 2$ . While any  $p > 0$  ensures that the matrix  $L$  approximates  $A$ , the choice of  $p$  determines how the *non-uniformity* in approximation error is penalized. For example, an  $\ell_1$  penalty allows some errors  $|A_{ij} - L_{ij}|$  to be much larger than others (as long

---

<sup>\*</sup>Equal contribution <sup>1</sup>School of Computing, University of Utah, Salt Lake City, Utah, USA. Correspondence to: Maheshakya <pmaheshakya4@gmail.com>.

as the total sum is small), while as  $p \rightarrow \infty$ , higher errors are penalized severely. Thus small values of  $p$  are used when some entries can be ignored as outliers (e.g., [Candes & Recht \(2008\)](#)), while higher values of  $p$  ensure a more uniform approximation.

The works of [Song et al. \(2017\)](#) and [Ban et al. \(2019a\)](#) develop sketching based algorithms for  $\ell_p$  norm approximation, particularly for  $p \in [1, 2]$ . They aim to find low-rank approximations whose objective value is  $\leq (1 + \epsilon)$  times the optimum. [Chierichetti et al. \(2017\)](#) develop approximation guarantees in much more generality, for all  $p \geq 1$  (including  $p = \infty$ ). Their result gives a simple  $O(k \log n)$  multiplicative approximation to the optimal error.

**Goals.** Our goal in this paper is to consider weighted low rank approximation with  $\ell_2$  and  $\ell_p$  error objectives and develop efficient and practical algorithms. We prove the efficacy of the greedy procedure under a novel yet natural assumption and establish additive error guarantees.

### 1.1. Our Results

In all our results, we assume that  $A$  is the input matrix, and that  $W$  is the non-negative weight matrix which has been re-scaled to satisfy  $W_{ij} \in [0, 1]$  for all  $i, j$ .

Our first result is to develop a simple greedy algorithm that gives an additive error approximation to weighted low-rank approximation. Unlike prior work, our analysis does not require any explicit assumptions on the weight matrix itself. It works as long as the *target matrix* (the intended low rank approximation) has a Frobenius norm not too large compared to  $A$ . Formally, our theorem is the following:

**Theorem 1.** *Suppose there exists a rank  $k$  matrix  $L$  that satisfies the two conditions: (a)  $\text{Cost}(L) \leq \Gamma$  and (b)  $\|L\|_F^2 \leq \Lambda \|A\|_F^2$  for some parameters  $\Lambda, \Gamma$ . Then for any  $\epsilon > 0$ , there exists an efficient algorithm that outputs a matrix  $L'$  of rank at most  $O(k\Lambda/\epsilon^2)$  that satisfies*

$$\text{Cost}(L') \leq \Gamma + \epsilon \|A\|_F^2.$$

*Remark.* Note that the guarantee is not in terms of the optimal error but in terms of  $\Gamma$ . This is because we could have the optimal matrix  $L^*$  having a large value of  $\|L^*\|_F$ , but there may exist an  $L$  with only a slightly larger cost, but a much smaller value of  $\|L\|_F^2$ .

We also note that in the unweighted case ( $W = \mathbf{1}_{d \times n}$ ), the bound on  $\|L\|_F / \|A\|_F$  is automatically satisfied: indeed, the ratio is always  $\leq 1$ . However in the weighted case, there can be pathological cases where a low-rank approximation has a much higher Frobenius norm than  $A$ . As an example, consider the case of  $A = W = \mathbf{I}_n$  ( $n \times n$  identity). The matrix  $L = \mathbf{1}_{n \times n}$  (all ones) is a rank-one matrix that achieves zero weighted approximation error. However,

we have  $\|L\|_F^2 / \|A\|_F^2 = n$ . Informally, our assumption is equivalent to requiring that even the “unimportant” entries in  $A$  are not too different in magnitude from the corresponding entries in  $L$ , on average. We believe that this is a reasonable assumption when approximating  $A$  by  $L$ . Moreover, under this assumption, the theorem requires *no structural assumptions* on  $W$  (as in prior work).

*Remark.* It is natural to ask if a dependence on  $\Lambda$  is *necessary* in general. Showing lower bounds in terms of this parameter is an interesting open direction. However, we note that the known hardness results for matrix completion give an evidence for hardness when  $A, W$  are sparse (in this case,  $\|L\|_F / \|A\|_F$  is  $\Theta(n)$ ). Specifically, [Hardt et al. \(2014\)](#) show that for matrix completion, given a matrix  $A$  which is the restriction to indices  $\Omega$  of a rank- $k$  matrix  $L$  with entries of magnitude  $O(1)$ , for any constant  $c$ , it is hard to construct a matrix  $B$  of rank  $r = ck$  such that  $\sum_{(i,j) \in \Omega} |A_{ij} - B_{ij}|^2 \leq \epsilon n$ . (This is assuming the hardness of an appropriate variant of coloring.) Viewing  $W$  as the binary mask matrix corresponding to  $\Omega$ , this also shows the hardness of weighted low rank approximation. The catch is that the amount of additive error allowed above is quite small; it is  $\epsilon \|A\|_F^2$  only when the matrix is sparse.

The algorithm is a greedy procedure that iteratively adds a rank 1 matrix to a decomposition, similar to Frank-Wolfe methods (see, e.g., [Clarkson \(2010\)](#)). The crux of the analysis is in showing that in spite of potentially bad choices in the past, there exists an update that can significantly improve the decomposition. A powerful feature of our techniques is that we can extend them to weighted approximation with  $\ell_p$  norm error. We study the entrywise  $\ell_p$  version of the objective in (1), defined as

$$\text{Cost}_p(L) = \sum_{i,j} W_{ij} \cdot |A_{ij} - L_{ij}|^p.$$

Here, additive error will correspond to an  $\ell_p$  analog of the Frobenius norm,  $\|X\|_{F_p} := (\sum_{i,j} |X_{ij}|^p)^{1/p}$ .

**Theorem 2.** *Let  $p > 2$ , and suppose there exists a rank  $k$  matrix  $L$  that satisfies: (a)  $\text{Cost}_p(L) \leq \Gamma$  and (b)  $\|L\|_{F_p}^p \leq \Lambda \|A\|_{F_p}^p$  for some parameters  $\Lambda, \Gamma$ . Then for any  $\epsilon > 0$ , there exists an efficient algorithm that outputs a matrix  $L'$  that satisfies  $\text{Cost}_p(L') \leq \Gamma + \epsilon \|A\|_{F_p}^p$ . Moreover the rank of  $L'$  is at most  $O\left(\frac{k^2 \Lambda^{2/p}}{\epsilon^{1+\frac{2}{p}}}\right)$ .*

We remark that this does not strictly dominate Theorem 1 because of the additional factor of  $k$ . For  $p \neq 2$ , this extra factor appears because the choice of basis for the target low-rank subspace is important to analyzing the greedy algorithm. As we discuss in Section 3, we need to use a carefully chosen basis for our argument.

Another remark is that our ideas only apply to  $p > 2$ . When

$p < 2$ , the maximization problem in each iteration of our current approach turns out to be that of computing the hypercontractive norm of a matrix, which is known to be hard (Barak et al., 2012; Ban et al., 2019a).

Our algorithm follows a similar outline as the one for Theorem 1, but it turns out to be much harder to analyze the improvement. We end up using tools from recent works on  $\ell_p$  regression (Adil et al., 2019; Bubeck et al., 2018). Moreover, finding a rank-one update in each step requires approximately computing the  $p \mapsto 2$  operator norm of an appropriate matrix, which can be done efficiently for  $p > 2$  using semidefinite programming, as shown by Nesterov (1998); Bhattiprolu et al. (2019).

Finally, as discussed in the introduction, even the unweighted version of low-rank approximation with entrywise  $\ell_p$  error has received a lot of interest, and is known to be challenging. Here, we obtain the following additive approximation.

**Theorem 3.** *Let  $p > 2$ , and let  $\text{OPT}_k$  denote the error of the best rank- $k$  approximation of a given matrix  $A$  in the entrywise  $\ell_p$  norm. There exists an efficient (polynomial time) algorithm that outputs an  $L'$  of rank  $O\left(\frac{k^2}{\epsilon^{1+\frac{2}{p}}}\right)$  that satisfies the error bound*

$$\|A - L'\|_{F_p}^p \leq \text{OPT}_k + \epsilon \|A\|_{F_p}^p.$$

Unlike the previous theorems, this result is *unconditional*. Indeed, it is a simple consequence of Theorem 2 (see Section 3.4). But to the best of our knowledge, such an additive error approximation for  $\ell_p$  low rank approximation was not known for  $p > 2$ . Given known hardness results for purely multiplicative approximation, it is interesting to study additive error guarantees (see Ban et al. (2019a)).

Our algorithm for Theorem 3 can be viewed as extending the familiar *iterative peeling* algorithm for  $\ell_2$  low-rank approximation to the  $\ell_p$  setting, for  $p > 2$ . The iterative step is different (now involving a  $p \mapsto 2$  norm computation), and we obtain an additive error guarantee. The theorem also complements the sketching-based algorithms for obtaining bi-criteria algorithms for  $p \in [1, 2)$  from Ban et al. (2019a). Finally, note that when the optimal error  $\text{OPT}_k$  is very small  $\ll \frac{\epsilon}{k \log n} \|A\|_{F_p}^p$ , the algorithm of Chierichetti et al. (2017) has a better guarantee than Theorem 3.

## 1.2. Notation and Overview

All the matrix and vector notations used in the paper will be defined at first use. We begin in Section 2 with the greedy algorithm for the weighted Frobenius error. The framework is then extended to the case of weighted  $\ell_p$  norm error in Section 3. The case of unweighted  $\ell_p$  error (Theorem 3) follows as a corollary and is presented in Section 3.4.

## 2. Algorithm for Squared Error

We now present the greedy framework that underlies all of our algorithms.

**Outline.** Our algorithm proceeds by maintaining a low-rank approximation for  $A$  and iteratively adding a rank-1 component that ensures sufficient error reduction. This is done by finding a vector  $\mathbf{z}$  and subtracting an appropriate multiple of  $\mathbf{z}$  from the *residuals* of each column. The analysis proceeds in a column-by-column fashion, and thus we begin with a few useful lemmas about approximating a single column, and present Algorithm 1 and its analysis in Section 2.1.

Our analysis is similar in spirit to the analysis of the greedy algorithm for column subset selection and sparse coding, (Altschuler et al., 2016; Bhaskara & Tai, 2019), but we need a different view in order to incorporate weights for entries. We begin with a few lemmas about approximating a single column using a collection of vectors. Let  $a \in \mathbb{R}^d$  be a vector, and  $w \in \mathbb{R}^d$  be weights for the coordinates. Define the function  $f_w : \mathbb{R}^d \mapsto \mathbb{R}$  as:

$$f_w(v) = \sum_{i \in [d]} w_i (a_i - v_i)^2, \quad (2)$$

where  $w_i, a_i, v_i$  denote the  $i$ th entries of the corresponding vectors. Next, suppose that  $x$  is a vector (which will be our current approximation for  $a$ ). Assume that  $x$  is “locally optimal” in the sense that increasing or decreasing the magnitude of  $x$  does not reduce the value of  $f_w$ . Formally,  $x$  satisfies  $\langle \nabla f_w(x), x \rangle = 0$ . The gradient has a simple form in our setting,  $\nabla f_w(v) = 2w \circ (a - v)$  (recall that  $\circ$  denotes the Hadamard or element-wise product). The following lemma shows how moving along a certain direction improves the value of  $f_w$ . First, we define

$$g_w(x, u) = \min_{\eta} f_w(x - \eta u), \quad (3)$$

which is the least possible value of  $f_w$  that can be obtained by moving from  $x$  along the direction  $u$ . (As we can set  $\eta = 0$ ,  $g_w(x, u)$  is always  $\leq f_w(x)$ .)

**Lemma 4.** *Let  $a, x, w$  be defined as above, and let  $u \in \mathbb{R}^d$  be a vector such that  $|\langle \nabla f_w(x), u \rangle| \geq \gamma$  and  $\sum_i w_i u_i^2 \leq 1$ . Then we have*

$$g_w(x, u) \leq f_w(x) - \frac{\gamma^2}{4}.$$

The proof follows by a direct expansion of  $f_w$ , and we defer the details to the supplement.

Next, we show a lemma that is central to our argument. It says that if there is some  $u$  such that  $f_w(u) < f_w(x)$ , and if  $u$  can be written as a linear combination of some basis vectors using “small” coefficients, then one of the basis directions can lead to a sufficiently large reduction in the value of  $f_w$ . Formally,

**Lemma 5.** Let  $u_1, u_2, \dots, u_k \in \mathbb{R}^d$  be arbitrary vectors, and suppose  $u = \sum_j \alpha_j u_j$ , where  $\sum_j \alpha_j^2 = B$ . Let  $a, x, w$  be defined as above, and suppose that  $f_w(u) < f_w(x)$ . Then

$$\sum_{j=1}^k |\langle \nabla f_w(x), u_j \rangle|^2 \geq \frac{(f_w(x) - f_w(u))^2}{B}.$$

*Proof.* We first observe that because of the convexity of  $f_w$  (it is a non-negative sum of convex functions), we have that

$$f_w(u) \geq f_w(x) + \langle \nabla f_w(x), u - x \rangle = f_w(x) + \langle \nabla f_w(x), u \rangle.$$

The last equality is because of our assumption that scaling  $x$  will not improve  $f_w$ . Because  $f_w(u) < f_w(x)$ , this implies that  $|\langle \nabla f_w(x), u \rangle| \geq f_w(x) - f_w(u)$ . Now, plugging in  $u = \sum_j \alpha_j u_j$  and applying Cauchy-Schwartz, we obtain:

$$\left( \sum_j \alpha_j^2 \right) \left( \sum_j |\langle \nabla f_w(x), u_j \rangle|^2 \right) \geq (f_w(x) - f_w(u))^2.$$

The first term is  $B$  by definition, and this completes the proof of the lemma.  $\square$

## 2.1. Algorithm for Weighted Approximation

The algorithm proceeds as follows: at time step  $t = 0, 1, \dots$ , an approximation  $\mathbf{x}_j^{(t)}$  is maintained for every column  $\mathbf{a}_j$ . Unlike in the single column case above, we now have (potentially) different weight vectors  $\mathbf{w}_j$  for each column  $j$ . We thus define

$$f_j(v) = \sum_{r \in [d]} w_{j,r} (a_{j,r} - v_r)^2, \quad (4)$$

where  $w_{j,r}$  denotes the  $r$ th coordinate of  $\mathbf{w}_j$  (similarly for  $\mathbf{a}_j$ ). Since our goal is an additive error approximation, an ideal goal is to bring  $f_j(v)$  within  $\epsilon \|\mathbf{a}_j\|_2^2$  of the optimal approximation for column  $j$ , for all  $j$ . (Algorithm 1 gives a full description of the procedure.)

**Remark.** Steps 7 and 8 of the algorithm involve a line search. This is easy in our case because the associated functions of  $\eta$  are univariate quadratics.

We start with some notation concerning the target rank- $k$  solution  $L$  (as promised by the statement of Theorem 1). Suppose that  $L = UV^T$ , where the columns of  $U$  are orthonormal, and let  $\mathbf{u}_j \in \mathbb{R}^d, \mathbf{v}_j \in \mathbb{R}^k$  denote the  $j$ th columns of  $U$  and  $V^T$  respectively. Because of the orthonormal columns in  $U$ , we have  $\|\mathbf{v}_j\|_2 = \|L_j\|_2$ , where  $L_j$  is the  $j$ th column of  $L$ . Our first goal is to obtain a column-wise control on  $\|L_j\| / \|\mathbf{a}_j\|$ . Define the column  $j$  to be *good* if  $\|L_j\|^2 / \|\mathbf{a}_j\|^2 \leq \Lambda/\epsilon$  and *bad* otherwise. In what follows, we denote by  $\mathcal{G}$  the set of all good columns. The following lemma is easy to see.

---

## Algorithm 1 Weighted low rank approximation with $L_2$ error

---

- 1: **Input:** Matrix  $A \in \mathbb{R}^{d \times n}$ , error parameter  $\epsilon$
  - 2: **Output:** Low-rank approximation  $L' \in \mathbb{R}^{d \times n}$  whose columns are spanned by a set of vectors  $Z$ , with  $|Z| = k' := 8k\Lambda/\epsilon^2$ .
  - 3: Initialize  $Z = \emptyset$ , set  $\mathbf{x}_j^{(0)} = 0$  for all  $j$
  - 4: **for**  $t = 1, 2, \dots, k'$  **do**
  - 5:   Using  $f_j$  defined in (4), let  $\mathbf{z} \in \mathbb{R}^d, \|\mathbf{z}\|_2 = 1$  be the vector that maximizes  $\sum_j \langle \nabla f_j(\mathbf{x}_j^{(t-1)}), \mathbf{z} \rangle^2$ , and add  $\mathbf{z}$  to  $Z$
  - 6:   **for each**  $j \in [n]$  **do**
  - 7:     Compute  $\eta$  that minimizes  $f_j(\mathbf{x}_j^{(t-1)} + \eta \mathbf{z})$ , and set  $\mathbf{x}' = \mathbf{x}_j^{(t-1)} + \eta \mathbf{z}$
  - 8:     Compute  $\eta$  that minimizes  $f_j(\eta \mathbf{x}')$  and set  $\mathbf{x}_j^{(t)} = \eta \mathbf{x}'$
  - 9:   **end for**
  - 10: **end for**
  - 11: Return  $Z$  and the associated low rank approximation  $L'$
- 

**Lemma 6.** The total mass of the bad columns of  $A$  is small. I.e.,  $\sum_{j \notin \mathcal{G}} \|\mathbf{a}_j\|^2 \leq \epsilon \|A\|_F^2$ .

*Proof.* Suppose the contrary, and assume that the inequality fails to hold. By the definition of bad, we have that

$$\sum_{j \notin \mathcal{G}} \|L_j\|^2 > \sum_{j \notin \mathcal{G}} \frac{\Lambda}{\epsilon} \|\mathbf{a}_j\|^2 \geq \Lambda \|A\|_F^2.$$

This contradicts our assumption about the bound on  $\|L\|_F^2$  (property (b) in Theorem 1).  $\square$

The lemma allows us to focus on the good columns for most of our analysis. We now introduce the following notation to track the progress of the algorithm.

**Notation.** We denote

$$\delta_j = \frac{f_j(L_j)}{\|\mathbf{a}_j\|_2^2}, \quad \theta_j^{(t)} = \frac{f_j(\mathbf{x}_j^{(t)})}{\|\mathbf{a}_j\|_2^2}. \quad (5)$$

Thus, informally, our goal is to ensure that  $\theta_j^{(t)} \geq \delta_j - \epsilon$  on average. We also study the following weighted averages:

$$\delta^* = \frac{\sum_{j \in \mathcal{G}} \|\mathbf{a}_j\|_2^2 \delta_j}{\|A_{\mathcal{G}}\|_F^2}, \quad \psi^{(t)} = \frac{\sum_{j \in \mathcal{G}} \|\mathbf{a}_j\|_2^2 \theta_j^{(t)}}{\|A_{\mathcal{G}}\|_F^2}, \quad (6)$$

where  $A_{\mathcal{G}}$  is the submatrix of  $A$  comprising only the good columns. The next lemma shows that if  $\psi^{(t)} - \delta^*$  is large, then the  $(t+1)$ th iteration makes considerable progress. Formally,

**Lemma 7.** Suppose that after the  $t$ 'th iteration of the algorithm we have  $\psi^{(t)} > \delta^*$ . Then there exists a  $\mathbf{z}$  such that

$$\sum_{j \in \mathcal{G}} |\langle \nabla f_j(\mathbf{x}_j^{(t)}), \mathbf{z} \rangle|^2 \geq \frac{\epsilon \|A_{\mathcal{G}}\|_F^2 (\psi^{(t)} - \delta^*)^2}{k\Lambda}$$

*Proof.* The idea will be to prove that one of the  $\{\mathbf{u}_i\}_{i \in [k]}$  satisfies the condition of the lemma. We do this by applying Lemma 5 to each of the good columns. Step 8 ensures that the current representation for each column cannot be improved by rescaling, which is essential for applying Lemma 5. Consider any  $j \in \mathcal{G}$ . This implies that  $L_j$  can be written as  $\sum_{i \in [k]} \alpha_i \mathbf{u}_i$ , where  $\sum_i \alpha_i^2 \leq \frac{\epsilon}{\Lambda} \|\mathbf{a}_j\|_2^2$  (indeed the  $\alpha_i$  are precisely the entries of the column  $\mathbf{v}_j$ ). Thus, by applying Lemma 5, we get

$$\begin{aligned} \sum_{i \in [k]} |\langle \nabla f_j(\mathbf{x}_j^{(t)}), \mathbf{u}_i \rangle|^2 &\geq \frac{\epsilon (f_j(\mathbf{x}_j^{(t)}) - f_j(L_j))^2}{\Lambda \|\mathbf{a}_j\|_2^2} \\ &= \frac{\epsilon \|\mathbf{a}_j\|_2^2 (\theta_j^{(t)} - \delta_j)_+^2}{\Lambda}, \end{aligned} \quad (7)$$

where  $(x)_+ = \max(x, 0)$ .

We first show that the sum of the RHS above over  $j \in \mathcal{G}$  is large. By viewing  $\frac{\|\mathbf{a}_j\|_2^2}{\|A_{\mathcal{G}}\|_F^2}$  as a probability distribution over the indices  $j \in \mathcal{G}$  and using the fact that  $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$ , we get

$$\sum_{j \in \mathcal{G}} \frac{\|\mathbf{a}_j\|_2^2}{\|A_{\mathcal{G}}\|_F^2} (\theta_j^{(t)} - \delta_j)_+^2 \geq \left( \sum_{j \in \mathcal{G}} \frac{\|\mathbf{a}_j\|_2^2}{\|A_{\mathcal{G}}\|_F^2} (\theta_j^{(t)} - \delta_j)_+ \right)^2$$

Using the observation that for any real numbers  $c, d$ ,  $(c)_+ + (d)_+ \geq (c + d)_+$  (and generalizing this to a sum of multiple terms), the RHS above can be simplified (using (6)) as

$$\sum_{j \in \mathcal{G}} \frac{\|\mathbf{a}_j\|_2^2}{\|A_{\mathcal{G}}\|_F^2} (\theta_j^{(t)} - \delta_j)_+ \geq (\psi^{(t)} - \delta^*)_+.$$

The RHS is positive by assumption, and thus plugging the above back into (7), we get:

$$\sum_{i \in [k]} \sum_{j \in \mathcal{G}} |\langle \nabla f_j(\mathbf{x}_j^{(t)}), \mathbf{u}_i \rangle|^2 \geq \frac{\epsilon \|A_{\mathcal{G}}\|_F^2 (\psi^{(t)} - \delta^*)_+^2}{\Lambda}.$$

Thus, by averaging, there exists an index  $i$  that satisfies the conclusion of the lemma. This completes the proof.  $\square$

The next lemma bounds the progress after  $t$  steps of the algorithm.

**Lemma 8.** *Let  $\epsilon < 1/2$  be a given error parameter. The number of iterations needed to achieve  $\psi^{(t)} - \delta^* \leq 2\epsilon$  is  $O\left(\frac{k\Lambda}{\epsilon^2}\right)$ .*

*Proof.* Recall that  $\psi^{(t)}$  and  $\delta^*$  only involve the good columns. Define  $\beta_t := \psi^{(t)} - \delta^*$ , and note that  $\beta_t$  clearly only reduces as  $t$  increases. We are done if  $\beta_t \leq 2\epsilon$ , and thus consider some  $t \leq \frac{8k\Lambda}{\epsilon^2}$  and assume that  $\beta_t > 2\epsilon$ .

We claim that in the next  $O(k\Lambda/\epsilon\beta_t)$  steps, the value of  $\beta_t$  reduces by a factor 2. To see this, suppose the contrary.

Now in each iteration, the algorithm finds some  $\mathbf{z}$  with  $\|\mathbf{z}\| = 1$  such that the total leftover mass (over *all* the columns) reduces by at least the bound given by Lemma 7. This is because the algorithm finds  $\mathbf{z}$  that maximizes  $\sum_j \langle \nabla f_j(\mathbf{x}_j^{(t-1)}), \mathbf{z} \rangle^2$ , and by Lemma 4, this also quantifies the total mass reduction. (Note that we have used the fact that all the weights are  $\in [0, 1]$  when applying the Lemma.) Thus, since  $\beta_{t'} \geq \beta_t/2$  for all the time steps  $t'$  we are considering, the mass reduction is at least

$$\frac{\epsilon \|A_{\mathcal{G}}\|_F^2 \beta_t^2}{4k\Lambda} \geq \frac{\epsilon \|A\|_F^2 \beta_t^2}{8k\Lambda},$$

where we used  $\epsilon < 1/2$  and Lemma 6. Thus if this continues for  $8k\Lambda/\epsilon\beta_t$  steps, the total mass reduction (which includes the reduction on bad columns) is  $\geq \beta_t \|A\|_F^2$ . But since  $\beta_t > 2\epsilon$  and at most  $\epsilon \|A\|_F^2$  of the mass is on the bad columns, this contradicts our assumption that  $\beta_t$  did not reduce by a factor 2.

Thus, we have argued that as long as  $\beta_t > 2\epsilon$ , it takes  $\leq 8k\Lambda/\epsilon\beta_t$  steps for  $\beta_t$  to reduce to  $\beta_t/2$ . Since  $\beta_0 \leq 1$ , we have that it takes  $\leq 2^j \cdot \frac{8k\Lambda}{\epsilon}$  steps for  $\beta_t$  to reduce from  $2^{-j}$  to  $2^{-(j+1)}$ . Thus, as the geometric series converges to twice the last term, we have that  $\beta_t$  reduces to  $\leq 2\epsilon$  after  $\frac{2}{2\epsilon} \cdot \frac{8k\Lambda}{\epsilon}$  steps, completing the proof of the lemma.  $\square$

We can now complete the proof of Theorem 1.

*Proof of Theorem 1.* Lemma 8 gives us that after  $O\left(\frac{k\Lambda}{\epsilon^2}\right)$  steps, we have  $\psi^{(t)} - \delta^* \leq 2\epsilon$ . Combined with Lemma 6, we have that the overall error in approximation is at most  $2\epsilon \|A_{\mathcal{G}}\|_F^2 + \epsilon \|A\|_F^2 \leq 3\epsilon \|A\|_F^2$ . This completes the proof (after replacing  $\epsilon$  by  $\epsilon/3$  throughout).  $\square$

### 3. Low Rank Approximation with $\ell_p$ Error

The high level outline of our algorithm is similar to the  $\ell_2$  setting. However, we need the right target decomposition, and need to set up the analysis carefully so as to make the rank-one update at every step efficient.

#### 3.1. Identifying a Target Decomposition

Let  $A$  be the matrix to be approximated and  $W$  the weight matrix as before. We make the same assumption:  $L = UV^T$  is the target decomposition, and we have  $\|L\|_{F_p}^p / \|A\|_{F_p}^p \leq \Lambda$ , for some parameter  $\Lambda$ .

Recall that the starting point in our analysis in the case of  $\ell_2$  error was to decompose  $L$  as  $UV^T$  using the SVD, so that we have a  $U$  with orthonormal columns, and a  $V$  such that  $\|L_j\| = \|\mathbf{v}_j\|$ . Implicit here is the fact that the  $\ell_2$  norm

is rotation invariant (using a different basis  $U$  maintains the norm property). Unfortunately, this is not true in the case of  $\ell_p$  norms. A priori, it is not clear if there exists a good decomposition that allows a property such as the above for all the columns, nor is it clear what normalization one should choose for the columns of  $U$ . E.g., should they have  $\|\cdot\|_p = 1$ , or a different norm such as  $\ell_2$  or the dual of  $\ell_p$ ?

So our first step is to describe the target decomposition and its properties.

**Lemma 9.** *Let  $L \in \mathbb{R}^{d \times n}$  be any rank  $k$  matrix with  $k \leq \min\{d, n\}$ . Then there exists a decomposition  $L = UV^T$  into  $(d \times k)$  and  $(k \times n)$  matrices such that (a) the columns of  $U$  satisfy  $\|\mathbf{u}_i\|_p = 1$  for all  $i \in [k]$ , (b) for all  $j \in [n]$ , the columns of  $V^T$  satisfy  $\|\mathbf{v}_j\|_\infty \leq \|L_j\|_p$ .*

*Proof.* The proof uses the following simple observation about rank  $k$  matrices.

*Observation.* Let  $M \in \mathbb{R}^{d \times n}$  be a rank  $k$  matrix. Then there exist a subset  $S$  of  $k$  columns of  $M$  with the property that all the other columns can be expressed as  $\sum_{i \in S} \alpha_i M_i$ , with  $|\alpha_i| \leq 1$  for all  $i$ .

The observation follows by an extremal argument, considering the  $k$  columns such that the volume of the associated parallelepiped is maximized. We refer the reader to [Chierichetti et al. \(2017\)](#) (Lemma 2) for a proof. (The argument itself is classic, and the property above is related to the notion of an Auerbach basis in Banach spaces. See [Taylor \(1947\)](#); [Martini et al. \(2001\)](#)). One of the early applications of this idea in the CS literature was in the work of [Awerbuch & Kleinberg \(2004\)](#).)

For our lemma, we apply the observation above to the matrix  $M$  whose columns are  $M_j = \frac{L_j}{\|L_j\|_p}$ . Let the chosen columns of  $M$  be denoted by the vectors  $\mathbf{u}_i$ , for  $i \in [k]$ . Then we have that all the other  $M_j$  can be expressed as  $\sum_i \alpha_i \mathbf{u}_i$  with  $|\alpha_i| \leq 1$ , and thus the corresponding  $L_j$  can be expressed using coefficients  $|\alpha_i| \leq \|L_j\|_p$ . By construction, the  $\mathbf{u}_i$  have  $\|\cdot\|_p = 1$ , which completes the proof of the lemma.  $\square$

The lemma allows us to use the framework from Section 2 to develop an iterative algorithm.

### 3.2. Single Vector Analysis

The first main step is to obtain analogs of Lemmas 4 and 5

Let  $w, a \in \mathbb{R}^d$ , and define the function  $f_{w,p} : \mathbb{R}^d \mapsto \mathbb{R}$  as:

$$f_{w,p}(v) = \sum_{i \in [d]} w_i |a_i - v_i|^p, \quad (8)$$

where  $w_i, a_i$  are the  $i$ th entries as before. Observe that the function  $f_{w,p}$  is convex (as it is a sum of convex functions).

The key to our proof is an appropriate smoothness property for  $f$ , which we prove use the following lemma from [Adil et al. \(2019\)](#).

**Lemma 10** (Lemma 4.5 of [Adil et al. \(2019\)](#)). *Let  $y \in \mathbb{R}$  and  $\Delta$  be any parameter. Then we have*

$$|y + \Delta|^p \leq |y|^p + g\Delta + 2^p \gamma_p(|y|, \Delta),$$

where  $g$  is the derivative of  $|y|^p$ , i.e.,  $g = p|y|^{p-2}y$ , and  $\gamma_p$  is the function (originally introduced in [Bubeck et al. \(2018\)](#)):

$$\gamma_p(t, \Delta) = \begin{cases} \frac{p}{2} t^{p-2} \Delta^2 & \text{if } |\Delta| \leq t, \\ |\Delta|^p + \left(\frac{p}{2} - 1\right) t^p & \text{otherwise.} \end{cases}$$

Using this lemma, we will be able to show the following analog of Lemma 4.

**Lemma 11.** *Let  $a, x, w$  be defined as above, and let  $u \in \mathbb{R}^d$  be a vector such that  $|\langle \nabla f_{w,p}(x), u \rangle| \geq \gamma \geq 0$  and  $\sum_i w_i |u_i|^p \leq 1$ . Then there exists  $\eta$  such that for some constant  $c_p = O_p(1)$ ,*

$$f_{w,p}(x - \eta u) \leq f_{w,p}(x) - \frac{\gamma^2}{c_p (f_{w,p}(x))^{\frac{p-2}{p}}}.$$

The proof of the lemma relies on Lemma 10 and a careful analysis. We thus defer it to the supplement. Our analysis will also need an analog of Lemma 5 where  $f_w$  is replaced by  $f_{w,p}$ . This is immediate because the proof only relies on the convexity of  $f_w$ , and thus also applies to  $f_{w,p}$ .

### 3.3. Algorithm and Analysis

Similar to the  $\ell_2$  case, we define

$$f_{j,p}(v) = \sum_{r \in [d]} w_{j,r} |a_{j,r} - v_r|^p, \quad (9)$$

where  $w_j$  is the weight vector for the  $j$ th column and  $w_{j,r}$  denotes the  $r$ th coordinate of  $w_j$  (similarly for  $a_j$ ).

**Algorithm.** The algorithm for the  $\ell_p$  error case is precisely the same as before, but instead of working with the functions  $f_j$ , we work with  $f_{j,p}$  (when taking gradients). The main change is in Step 5 of the algorithm, where instead of finding a vector  $\mathbf{z}$  that maximizes  $\sum_j \langle \nabla f_j(\mathbf{x}_j^{(t)}), \mathbf{z} \rangle^2$  subject to  $\|\mathbf{z}\| = 1$  (which reduces to finding the top singular vector of an appropriate matrix), we now need to solve the following:

$$\max \sum_j \frac{\langle \nabla f_{j,p}(\mathbf{x}_j^{(t)}), \mathbf{z} \rangle^2}{(f_{j,p}(\mathbf{x}_j^{(t)}))^{\frac{p-2}{p}}} \text{ subject to } \|\mathbf{z}\|_p = 1. \quad (10)$$

This can be re-written as finding a vector  $\mathbf{z}$  that maximizes  $\|M\mathbf{z}\|_2^2$  subject to  $\|\mathbf{z}\|_p = 1$ , for an appropriate matrix  $M$

(which we can construct since we know  $\mathbf{x}_j^{(t)}$  and  $f$ ). This is exactly the problem of computing the so-called  $p \mapsto 2$  operator norm of the matrix  $M$ . The classic result of [Nesterov \(1998\)](#) shows that the problem admits a constant factor approximation. More recently, the work [Bhattiprolu et al. \(2019\)](#) obtains nearly tight factors for the problem. Both these algorithms are based on a semidefinite programming relaxation for approximating the operator norm, and crucially rely on  $p \geq 2$  in their analysis. We summarize these results as follows.

**Theorem 12.** ([Nesterov, 1998](#); [Bhattiprolu et al., 2019](#)) *For any  $p \geq 2$ , there exists an efficient (polynomial time) algorithm for approximating the  $p \mapsto 2$  operator norm of a matrix  $M$  to a factor only depending on  $p$  (which indeed turns out to be  $O(\sqrt{p})$  using the result of [Steinberg \(2005\)](#)) Specifically, the algorithm outputs a  $\mathbf{z}$  with  $\|\mathbf{z}\|_p = 1$ , such that the objective value in (10) is  $\Omega(\frac{1}{p})$  times the optimum.*

Our analysis once again involves quantities  $\delta_j$  and  $\theta_j^{(t)}$ , defined as follows:

$$\delta_j = \frac{f_{j,p}(L_j)}{\|\mathbf{a}_j\|_p^p}, \quad \theta_j^{(t)} = \frac{f_{j,p}(\mathbf{x}_j^{(t)})}{\|\mathbf{a}_j\|_p^p}. \quad (11)$$

We also define weighted averages as before:

$$\delta^* = \frac{\sum_{j \in \mathcal{G}} \|\mathbf{a}_j\|_p^p \delta_j}{\|A_{\mathcal{G}}\|_{F_p}^p}, \quad \psi^{(t)} = \frac{\sum_{j \in \mathcal{G}} \|\mathbf{a}_j\|_p^p \theta_j^{(t)}}{\|A_{\mathcal{G}}\|_{F_p}^p}. \quad (12)$$

The following lemma shows that as long as  $\psi^{(t)} - \delta^*$  is large enough, the algorithm makes significant progress.

**Lemma 13.** *Suppose that after the  $t$ 'th iteration of the algorithm we have  $\psi^{(t)} > \delta^*$ . Then there exists a unit vector  $\mathbf{z}$  such that*

$$\sum_{j \in \mathcal{G}} \frac{|\langle \nabla f_{j,p}(\mathbf{x}_j^{(t)}), \mathbf{z} \rangle|^2}{(f_{j,p}(\mathbf{x}_j^{(t)}))^{\frac{p-2}{p}}} \geq \frac{\epsilon^{2/p} \|A_{\mathcal{G}}\|_{F_p}^p (\psi^{(t)} - \delta^*)^2}{k^2 \Lambda^{2/p}}.$$

*Proof.* The proof follows the structure of that of [Lemma 7](#), and will show that one of the  $\mathbf{u}_i$  satisfy the conclusion of the lemma. Consider some good column  $j$ .

Our updates ensure that we can apply [Lemma 5](#) (where  $f_j$  is replaced by  $f_{j,p}$ ). The value of  $\Lambda$  that we use in the lemma statement is the following: every coefficient used is  $\leq \|L_j\|_p$  in magnitude, from [Lemma 9](#). Since  $j$  is a good column, this is at most  $(\frac{\Lambda}{\epsilon})^{1/p} \|\mathbf{a}_j\|_p$ . As there are  $k$  terms, the sum of squared coefficients is bounded by  $k \|\mathbf{a}_j\|_p^2 (\frac{\Lambda}{\epsilon})^{2/p}$ . Plugging this in, and writing  $C = (\frac{\Lambda}{\epsilon})^{2/p}$

for convenience, we obtain:

$$\begin{aligned} \sum_{i \in [k]} |\langle \nabla f_{j,p}(\mathbf{x}_j^{(t)}), \mathbf{u}_i \rangle|^2 &\geq \frac{(f_{j,p}(\mathbf{x}_j^{(t)}) - f_{j,p}(L_j))_+^2}{kC \|\mathbf{a}_j\|_p^2} \\ &= \frac{\|\mathbf{a}_j\|_p^{2p} (\theta_j^{(t)} - \delta_j)_+^2}{kC \|\mathbf{a}_j\|_p^2}. \end{aligned} \quad (13)$$

Thus, since  $f_{j,p}(\mathbf{x}_j^{(t)}) = \theta_j^{(t)} \|\mathbf{a}_j^{(t)}\|_p^p$  by definition, we have (after plugging in above and simplifying the exponent of  $\|\mathbf{a}_j\|_p$ )

$$\begin{aligned} \sum_{i \in [k]} \frac{|\langle \nabla f_{j,p}(\mathbf{x}_j^{(t)}), \mathbf{u}_i \rangle|^2}{(f_{j,p}(\mathbf{x}_j^{(t)}))^{\frac{p-2}{p}}} &\geq \frac{\|\mathbf{a}_j\|_p^p (\theta_j^{(t)} - \delta_j)_+^2}{kC (\theta_j^{(t)})^{\frac{p-2}{p}}} \\ &\geq \frac{\|\mathbf{a}_j\|_p^p (\theta_j^{(t)} - \delta_j)_+^2}{kC}. \end{aligned}$$

The second inequality uses the fact that  $p \geq 2$  and  $\theta_j^{(t)} \in (0, 1]$ . Then, we can sum over the columns  $j \in \mathcal{G}$ , and mimicking the idea from the proof of [Lemma 7](#) (this time using  $\|\mathbf{a}_j\|_p^p / \|A_{\mathcal{G}}\|_{F_p}^p$  as the distribution), we get

$$\sum_{i \in [k]} \sum_{j \in \mathcal{G}} \frac{|\langle \nabla f_{j,p}(\mathbf{x}_j^{(t)}), \mathbf{u}_i \rangle|^2}{(f_{j,p}(\mathbf{x}_j^{(t)}))^{\frac{p-2}{p}}} \geq \frac{\|A_{\mathcal{G}}\|_{F_p}^p (\psi^{(t)} - \delta^*)^2}{kC}.$$

Thus by averaging and plugging in the value of  $C$ , one of the  $\mathbf{u}_i$  must satisfy the conclusion of the lemma.  $\square$

Given this lemma, the proof of [Theorem 2](#) follows as before.

*Proof of Theorem 2.* We first observe that an analog of [Lemma 8](#) also holds in the current setting, with two differences. The first is the change from  $\frac{\epsilon}{k\Lambda}$  to  $\frac{\epsilon^{2/p}}{k^2 \Lambda^{2/p}}$  in the coefficient for the improvement at each step. Secondly, we only solve the  $p \mapsto 2$  norm problem approximately, and this adds an additional factor of  $p$  to the bound. Thus, the number of steps needed for the  $\beta_t$  to halve is now  $O\left(\frac{pk^2 \Lambda^{2/p}}{\epsilon^{2/p}}\right)$ , and thus reducing to  $2\epsilon$  requires number of steps equal to the bound in the statement of the Theorem.

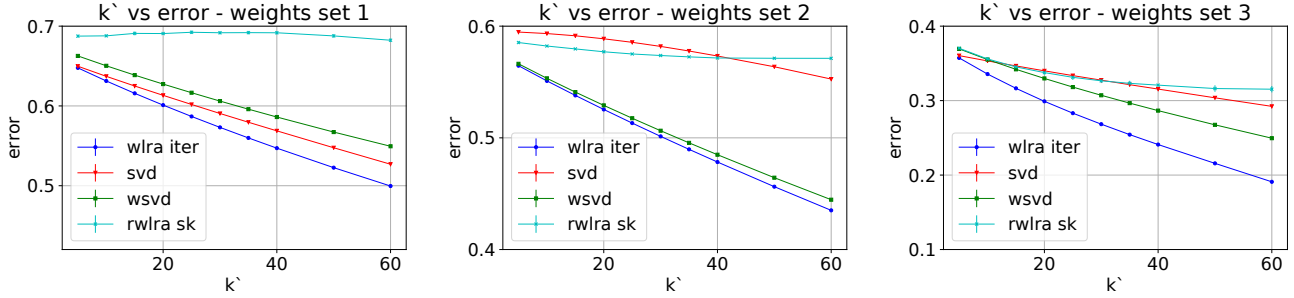
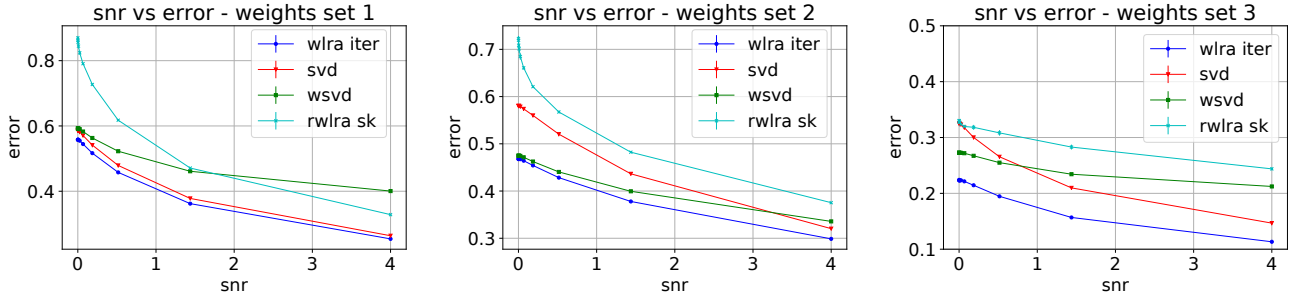
Finally, observing that the bound on the total mass of the bad columns carries over to the  $\ell_p$  case, the theorem follows.  $\square$

### 3.4. Unconditional Result for Uniform Weights

We now show how to deduce [Theorem 3](#) using [Theorem 2](#).

*Proof of Theorem 3.* We only need to check that the matrix achieving the optimal error (say  $L^*$ ) satisfies the conditions of [Theorem 2](#). This is true because

$$\|L^*\|_{F_p} = \|(L^* - A) + A\|_{F_p} \leq \|L^* - A\|_{F_p} + \|A\|_{F_p}.$$


 Figure 1. Error of *wltra-iter*, *svd*, *wsvd*, *rwlra-sk* as  $k'$  is increased.

 Figure 2. Error of *wltra-iter*, *svd*, *wsvd*, *rwlra-sk* as SNR is increased.

By definition,  $\|L^* - A\|_{F_p} = \text{OPT}_k$ , which is  $\leq \|A\|_{F_p}$ . This implies that the assumption holds with  $\Lambda = 2^p$ .  $\square$

## 4. Experiments

In this section we evaluate our algorithm (*wltra-iter*) for weighted low rank approximation by comparing its performance with three baselines: (a) applying SVD to the matrix  $A$  (*svd*) (b) applying SVD to weighted matrix  $W \circ A$  (*wsvd*) (c) regularized weighted low rank approximation algorithm with sketching in Ban et al. (2019b) (*rwlra-sk*). In (c), we use the alternating minimization based algorithm provided in Ban et al. (2019b). We present experiments on synthetic data below and defer other results to the supplement.

We conduct two sets of experiments. In the first set, we vary the output rank  $k'$  and show how the error changes for each algorithm. In the second set, we demonstrate how the error in each algorithm changes as the signal to noise ratio (SNR) varies: the signal is a low rank matrix and we add Gaussian noise to it. In each experiment, we measure the scaled error  $(\sum_{ij} W_{ij} (A_{ij} - Z_{ij})^2) / \|A\|_F^2$  where  $Z$  is the solution output by each algorithm (we note that in the experiments in Ban et al. (2019b), the objective value is plotted instead of the error thus our experiments are not comparable); we average results over 10 independent runs.

We first generate  $500 \times 5$  dimensional matrices  $M_1, M_2$  with random orthonormal vectors as columns and a diagonal matrix  $S$  with diagonal elements  $[1, 0.9, (0.9)^2, (0.9)^3, (0.9)^4]$

(normalized). Thus  $M = M_1 S M_2^T$  is a rank 5 matrix with  $\|M\|_F = 1$ . In each experiment we create matrix  $A$  by adding a noise matrix  $N$  with  $N_{ij} \sim \mathcal{N}(0, \sigma^2)$  to  $M$ . We set the sketch size parameter in *rwlra-sk* to 100 in all experiments. We generate weight matrices of  $500 \times 500$  dimension with the following configurations.

- $W_1$ : Each element is sampled from the interval  $[0, 1]$  uniformly at random.
- $W_2$ : Elements corresponding to largest 50000  $|A_{ij}|$ s are set to 0, and 1 elsewhere.
- $W_3$ : The random binary matrix is first chosen by setting each entry to 1 with probability 0.1 and 0 otherwise. Following this, the first 100 columns of first 150 rows are set to 1.

In a setting with random weights ( $W_1$ ), even simple baselines such as SVD achieve a small error as expected. However with semirandom weights (as in  $W_3$ ), SVD performs much more poorly compared to a greedy approach.

In the first set of experiments, we plot the error of each algorithm with output rank  $k'$  in the list (5, 10, 15, 20, 25, 30, 35, 40, 50, 60). Here we fix  $\sigma = 0.005$  (thus  $\text{SNR} \approx 0.16$ ) and  $\lambda = 0.05$  for weight matrix setting  $W_2$  and  $\lambda = 0.01$  for weight matrix settings  $W_1, W_3$  in *rwlra-wk*. Figure 1 shows the error of each algorithm for different weight matrices.

In the second set of experiments, we plot the error of each algorithm as the SNR is increased from 0.0004 to 4. Here



we fix  $k' = 50$  and  $\lambda = 0.005$  for weight matrix setting  $W_1$  and  $\lambda = 0.01$  for weight matrix settings  $W_2, W_3$  in *rwlra-wk*. We control SNR by changing  $\sigma$  appropriately. Figure 2 shows the error rates of each algorithm for different weight matrices. The results show the greedy procedure achieving small recovery error even in low SNR regimes.

In our supplement, we provide analogous results for four other choices of weight matrices as well as two real datasets. The full code is also provided.

## 5. Conclusion

We study a natural greedy algorithm for the weighted low rank approximation problem and establish novel additive error guarantees in  $\ell_2$  and  $\ell_p$  norms for  $p > 2$  under a new, realistic, assumption on the target low rank matrix. Our algorithm is easy to implement and works well in practice, compared to natural baselines and previous approaches.

## Acknowledgements

Aditya Bhaskara is partially supported by NSF CCF-2008688, an NSF CAREER Award (CCF-2047288) and by a Google Faculty Research Award.

## References

- Adil, D., Kyng, R., Peng, R., and Sachdeva, S. Iterative refinement for  $\ell_p$ -norm regression. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pp. 1405–1424, USA, 2019.
- Altschuler, J., Bhaskara, A., Fu, G., Mirrokni, V., Ros-tamizadeh, A., and Zadimoghaddam, M. Greedy column subset selection: New bounds and distributed algorithms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 2539–2548. JMLR.org, 2016.
- Awerbuch, B. and Kleinberg, R. D. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pp. 45–53, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138520. doi: 10.1145/1007352.1007367.
- Ban, F., Bhattiprolu, V., Bringmann, K., Kolev, P., Lee, E., and Woodruff, D. P. A ptas for  $\ell_p$ -low rank approximation. In Chan, T. M. (ed.), *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 747–766. SIAM, 2019a.
- Ban, F., Woodruff, D., and Zhang, R. Regularized weighted low rank approximation. *Advances in Neural Information Processing Systems*, 32:4059–4069, 2019b.
- Barak, B., Brandao, F. G., Harrow, A. W., Kelner, J., Steurer, D., and Zhou, Y. Hypercontractivity, sum-of-squares proofs, and their applications. *Proceedings of the 44th symposium on Theory of Computing - STOC '12*, 2012.
- Bhaskara, A. and Tai, W. M. Approximate Guarantees for Dictionary Learning. In Beygelzimer, A. and Hsu, D. (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 299–317, Phoenix, USA, 25–28 Jun 2019. PMLR. URL <http://proceedings.mlr.press/v99/bhaskara19a.html>.
- Bhattiprolu, V., Ghosh, M., Guruswami, V., Lee, E., and Tulsiani, M. Approximability of  $p \rightarrow q$  matrix norms: Generalized krivine rounding and hypercontractive hardness. In Chan, T. M. (ed.), *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 1358–1368. SIAM, 2019.
- Bubeck, S., Cohen, M. B., Lee, Y. T., and Li, Y. An homotopy method for  $\ell_p$  regression provably beyond self-concordance and in input-sparsity time. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pp. 1130–1137, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355599.
- Candes, E. J. and Recht, B. Exact matrix completion via convex optimization, 2008.
- Chierichetti, F., Gollapudi, S., Kumar, R., Lattanzi, S., Panigrahy, R., and Woodruff, D. P. Algorithms for  $\ell_p$  low-rank approximation. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 806–814, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- Clarkson, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4), September 2010. ISSN 1549-6325.
- Clarkson, K. L. and Woodruff, D. P. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6), January 2017. ISSN 0004-5411.
- Eriksson, A. and van den Hengel, A. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the  $\ell_1$  norm. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 771–778, 2010.

- Frieze, A., Kannan, R., and Vempala, S. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, November 2004.
- Gillis, N. and Glineur, F. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011.
- Hardt, M., Meka, R., Raghavendra, P., and Weitz, B. Computational limits for matrix completion. In Balcan, M. F., Feldman, V., and Szepesvári, C. (eds.), *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pp. 703–725, Barcelona, Spain, 13–15 Jun 2014. PMLR.
- Manton, J. H., Mahony, R., and Yingbo Hua. The geometry of weighted low-rank approximations. *IEEE Transactions on Signal Processing*, 51(2):500–514, 2003.
- Martini, H., Swanepoel, K. J., and Weiß, G. The geometry of minkowski spaces — a survey. part i. *Expositiones Mathematicae*, 19(2):97–142, 2001. ISSN 0723-0869.
- Musco, C., Musco, C., and Woodruff, D. P. Simple heuristics yield provable algorithms for masked low-rank approximation, 2020.
- Nesterov, Y. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9(1-3):141–160, 1998.
- Razenshteyn, I., Song, Z., and Woodruff, D. P. Weighted low rank approximations with provable guarantees. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC ’16, pp. 250–263, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341325.
- Song, Z., Woodruff, D. P., and Zhong, P. Low rank approximation with entrywise  $\ell_1$ -norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pp. 688–701, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450345286.
- Srebro, N. and Jaakkola, T. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML’03, pp. 720–727. AAAI Press, 2003. ISBN 1577351894.
- Steinberg, D. Computation of matrix norms with applications to robust optimization. *Research thesis, Technion-Israel University of Technology*, 2, 2005.
- Taylor, A. E. A geometric theorem and its application to biorthogonal systems. *Bulletin of the American Mathematical Society*, 53(6):614–616, 1947.
- Young, G. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6(1):49–53, Feb 1941. ISSN 1860-0980.