On Algorithms for Weighted Low Rank Approximation

Yucheng Dai

CMU-CS-23-115

May 2023

Thesis Committee:
David P. Woodruff (Chair)
Richard Peng

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Keywords: Weighted Low Rank Approximation, Sketching, Optimization, Sampling

Abstract

In this thesis we consider the Weighted Low Rank Approximation problem. For an $m \times n$ matrix A, we give a row-sampling algorithm algorithm running in $O(\text{nnz}(A) \log n + (m+n) \cdot \text{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A)))$ time, where all weights in the weight matrix are within a ratio of $1/w$, $\kappa(A)$ denotes the condition number of matrix A, and $\text{nnz}(A)$ denotes the number of non-zero entries of A. Our algorithm is bicriteria, and outputs a matrix of rank $O(\frac{k^2}{w^2\varepsilon^2\kappa^2(A)})$ and achieves a multiplicative $(1+\epsilon)$-approximation to the cost of the best rank-k matrix. Compared to prior work of Bhaskara et al. (PMLR, 2021), our algorithm achieves multiplicative rather than additive error, outputs a low rank approximation with right factor corresponding to an actual subset of rows of A, has leading order term in the running time of $\text{nnz}(A)$ rather than at least $\text{nnz}(A) \cdot k/\epsilon^2$, and has the same bicriteria rank in the worst case. We also show the $1/w$ factor is necessary in the bicriteria rank of any row subset selection algorithm.

Acknowledgments

My thesis committee is awesome!!
My thesis committee is awesome!!
My thesis committee is awesome!!
My thesis committee is awesome!!
My thesis committee is awesome!!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Low rank approximation is a fundamental problem in machine learning with numerous applications. It can be solved with the singular value decomposition, and there are also very efficient randomized approximation algorithms. For a survey, we refer the reader to [4, 12].

Unfortunately most variations of the low rank approximation problem are NP-hard. One well-studied variation is the weighted low rank approximation problem, for which the difficulty is first explained in [13]. The motivation for this problem is that in many applications, we might not want each entry to be weighted the same because some entries are less important. Instead, we would like to assign a non-negative weight to each entry of the matrix, for example, a probability or the reciprocal of the standard deviation of the distribution of values for that entry. This immediately breaks the linear algebraic structure of the matrix, which results in an NP-hard problem in general [7].

There are a number of ways of dealing with this NP-hardness. Efficient parameterized algorithms were presented in [10], assuming the weight matrix has a small number of distinct rows or columns, or has low rank. Several other works [1, 9] also assume the weight matrix has low rank or other nice structure; interestingly in [9] the output rank is related to the so-called communication complexity of the weight matrix. One can also deal with hardness by allowing the output to have larger rank than the desired rank k, which is called a bicriteria solution, and to also paramaterize the bicriteria rank in terms of the norm of near-optimal solutions - the larger the norm the larger the output rank. This was the method done in [2], which is most related to our work.

There are two natural notions of error: additive and multiplicative. For additive error, the goal is given a matrix $W \in (\mathbb{R} \setminus \mathbb{R}_-)^{m \times n}$ and $W \in \mathbb{R}^{m \times n}$, to find matrix $F \in \mathbb{R}^{m \times n}$ with low rank[1] such that:

[1]Not necessarily rank k

$$\|A - F\|_{W,F}^2 \leq \min_{M,\mathrm{rank}(M)=k} \|A - M\|_{W,F}^2 + \varepsilon \|A\|_{W,F}^2$$

For an $\varepsilon \|A\|_{W,F}^2$ additive error approximation. The notation here is defined in Definition 1.

And

$$\|A - F\|_{W,F}^2 \leq (1+\varepsilon) \min_{M,\mathrm{rank}(M)=k} \|A - M\|_{W,F}^2$$

For a $(1+\varepsilon)$ multiplicative error approximation.

The work of [2] in fact uses a weaker definition of additive error, which replaces the weighted norm $\|A\|_{W,F}^2$ with the unweighted version $\|A\|_F^2$. The latter is strictly weaker than the former.

Our Assumption. We consider weight matrices which have a positive minimum weight of w and a maximum weight of 1. By scaling, this captures any weight matrix with w being the ratio of the minimum to maximum weights. We do not have any other assumption on the weight matrix. We also assume the input matrix A has full rank, which can be achieved by an arbitrarily tiny random perturbation (at the cost of arbitrarily tiny additive error), and for many matrices in practice holds already because of measurement noise. Our assumption provides a type of regularization, in that it does not allow entries in the solution to be infinitely large, which could be possible with zero weights.

Our Results. We design a sampling algorithm which samples rows from matrix A. use the sampled rows as a right matrix and solve n weighted regression problems to obtain the left matrix. Here we cannot afford to solve these problems separately in our given time bound. Instead, we observe that the leverage scores of each of the m instances are within a factor of $1/w$ from each other so that oversampling by this factor allows for a single leverage score sampling matrix to solve each of the m weighted regression problems.

Multiplying our two factors together, we first show that with constant probability, the resulting matrix provides an additive $\varepsilon \|A\|_{W,F}^2$ weighted approximation if we sample $O(\frac{k^2}{w^2\varepsilon^2\kappa^2(A)})$ rows, where $\kappa(A)$ is the condition number of matrix A. In a number of cases, this algorithm gives a smaller bicriteria rank than the result of [2]. Moreover, this algorithm has a significantly smaller time complexity of $O(\mathrm{nnz}(A) + (m+n) \cdot \mathrm{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A))$.

Next, we use our additive error algorithm to construct a multiplicative error algorithm. Specifically, we first find a rank-$O(k)$ matrix providing an $O(1/w)$-approximation by performing standard unweighted Frobenius norm row subset selection, and then apply our previous additive error approximation algorithm on the residual. In this way we obtain a $(1+\varepsilon)$ approximation with a

matrix of rank at most $O(\frac{k^2}{w^4\varepsilon^2\kappa^2(A)})$ with constant probability and in $O(\text{nnz}(A)\log n + (m+n)\cdot \text{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A)))$ time.

Finally, we give lower bounds on the bicriteria rank of row sampling algorithms. Namely, in section 2.4, we show that we need to sample at least $\Omega(\frac{k}{w})$ rows for constant ϵ on certain inputs, namely, when W and A both look approximately like identity matrices[2].

Previous Work. The work of [2] studies additive error algorithms which iteratively selects the principle components of the residual (i.e., the best improvement in the unweighted case). We restate their algorithm as Algorithm 1 below.

This algorithm assumes a rank-k matrix L with the property that $\|L\|_F^2 \leq \Lambda \|A\|_F^2$ and shows that with $O(\frac{k\Lambda}{\varepsilon^2})$ rounds, the output achieves an additive error not much more than the cost of using L.

A sketch of their proof is as follows: consider only columns of L that constitute a significant portion of the Frobenius norm of L. One can show fast convergence on the cost function for these columns, while the remaining columns will not exceed the desired additive error bound.

Algorithm 1: Weighted Low Rank Approximation with Additive Error [2]

Input: Matrix $A \in \mathbb{R}^{d\times n}$, error parameter ε
Output: Low-rank Approximation $L' \in \mathbb{R}^{d\times n}$ whose columns are spanned by a set of vectors Z, with $|Z| = k' := 8k\Lambda/\varepsilon^2$.

Initialize $Z = \emptyset$, set $x_j^{(0)} = 0$ for all j.
for $t = 1 \cdots k'$ **do**
  Define $f_j(v) = \sum_{r\in[d]} W_{rj}(A_{rj} - v_r)^2$ for all j, let $z \in \mathbb{R}^d$, $\|z\|_2 = 1$ be the vector that maximizes $\sum_j \langle \nabla f_j(x_j^{(t-1)}, z)\rangle^2$, and add z to Z.
  for *each* $j \in [n]$ **do**
    Compute η that minimizes $f_j(x_j^{(t-1)} + \eta z)$ and set $x' = x_j^{(t-1)} + \eta z$.
    Computer η that minimizes $f_j(\eta x')$ and set $x_j^{(t)} = \eta x'$.
  end for
end for
Return Z and the associated low rank approximation L'.

With Low Rank Weight Matrix In [12], the author assumed that the weight matrix has rank r, and try to find a rank-k approximation instead of a rank-poly(k, ε) one. With this assumption, they utilized sketching technique and reduced the problem to a regression problem. This approach is very similar to ours, but as they pointed out in Assumption 1.3, there is no polyno-

[2]See detailed definition in Section 2.4

mial algorithm solving the problem. Their results showed that there exists an algorithm runs in $n^{O(k^2 r/\varepsilon)}$ time that gives $(1+\varepsilon)$ multiplicative error.

With Binary Weights. In [9], the author studied a variation where the weight matrix is binary. They proved that a simple unweighted low rank approximation to matrix $W \circ A$ (i.e. zero out entries of A where W is zero) can be a good enough additive error weighted approximation. Therefore, this algorithm runs in standard low rank approximation time, which is $O(\text{nnz}(A) + n \cdot \text{poly}(k'/\varepsilon))$ with k' be the rank of the approximation.

Our Techniques. Our techniques are completely different than those of [2]. Instead, we first compute a $O(1)$-approximate row sampling-based solution to the unweighted problem, which gives an $O(1/w)$-approximation to the weighted problem. We then show that adaptively sampling rows proportional to their squared distance to this approximation works.

Our analysis tries to mimic the standard proofs of adaptive sampling in the unweighted case, but the lack of a solution based on the singular value decomposition is a significant hindrance in our setting. The proof in the unweighted case shows that a special linear combination of the samples, depending on the t-th left singular vector of A, is close in expectation to the right singular vector of A, for each t, and then bounds the variance. Thus, the span of the samples can be shown to contain a projection which is "close enough" to the projection onto the top k singular vectors. In the weighted setting, since A has full row rank, we can still write k vectors in a basis for the rowspan of the optimal solution as a special linear combination of the rows of A.

Chapter 2

Main Results

2.1 Definitions

Definition 1. Given matrices $W \in (\mathbb{R} \setminus \mathbb{R}_-)^{m \times n}$ and $A \in \mathbb{R}^{m \times n}$, we define the weighted Frobenius norm as:

$$\|A\|_{W,F}^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij} A_{ij}^2$$

It is easy to show this is a norm.

Definition 2. We define a row sampling process similar to [6] as follows:

Given matrix $A \in \mathbb{R}^{m \times n}$ and $P_i \in [0, 1]$, $i = 1, \ldots, m$, with:

$$P_i \geq \alpha \frac{\left\|A^{(i)}\right\|^2}{\|A\|_F^2}$$

where $\alpha \leq 1$ is a constant (independent of m, n). If we set $\alpha = 1$, then P_i can be inferred from matrix A.

2.2 Weighted Low Rank Approximation with Additive Error

Theorem 3. *Given $\epsilon \in (0, 1)$ and a matrix $W \in \mathbb{R}^{m \times n}$ with entries in range $[w, 1]$ and $A \in \mathbb{R}^{m \times n}$ of full row rank, there exists an algorithm which finds matrix F of rank $O(\frac{k^2}{w^2 \varepsilon^2 \kappa^2(A)})$ such that*

$$\|A - F\|_{W,F}^2 \leq \left\|A - \hat{A}_k\right\|_{W,F}^2 + \varepsilon \|A\|_{W,F}^2$$

where $\hat{A}_k$ is the optimum rank-k weighted approximation and $\kappa(A)$ is the condition number of matrix A.

Proof. We use the row sampling procedure in Definition 2 to sample s i.i.d. rows of A with $\alpha = 1$. Denote the indices of rows sampled as a multiset S and let $s = |S|$.

Let $\hat{A}_k = TDV$ be the optimal weighted rank-k approximation, where $T \in \mathbb{R}^{m\times k}$ is orthonormal and $D \in \mathbb{R}^{k\times k}$ is a diagonal matrix satisfying $U^T A = V$ for matrix $U \in \mathbb{R}^{m\times k}$ whose rows have unit norm. Denote row t of U as $U^{(t)}$. We use similar notation for other matrices as well.

Notice that the diagonal entries of D is only related to singular values of $\hat{A}_k$ and A. Specifically, we can bound it by the following:

We know $\left\|DU^T A\right\|_F^2 = \left\|\hat{A}_k\right\|_F^2 \leq \frac{1}{w}\|A\|_F^2$. Let $A = U_A \Sigma V_A^T$, then we have:

$$\left\|DU^T U_A \Sigma\right\|_F^2 \leq \frac{1}{w}\sum_i \Sigma_{ii}^2$$

Notice that $U^T U_A$ is a projection onto a subspace, so the lower bound on the norm of each row of $U^T U_A \Sigma$ is the smallest singular value of A.

Hence, we derive: $\|D\|_F^2 = \sum_i D_{ii}^2 \leq \frac{k}{\kappa^2(A) w}$ where $\kappa(A)$ is the condition number of A.

Then, we would like to estimate the samples we selected. For $t = 1, \ldots, k$, we consider:

$$X^{(t)} = \frac{1}{s}\sum_{i\in S} \frac{U_i^{(t)}}{P_i} A^{(i)}$$

The expectation of $X^{(t)}$ is:

$$\mathbb{E}\left(X^{(t)}\right) = V^{(t)}$$

Therefore, plugging in $P_i = \frac{\left\|A^{(i)}\right\|^2}{\|A\|_F^2}$, an upper bound for the variance is given by:

$$\mathbb{E}\left\|V^{(t)} - X^{(t)}\right\|^2 \leq \frac{1}{s}\sum_{i=1}^{m} \frac{\left\|U_i^{(t)}\right\|^2 \left\|A^{(i)}\right\|^2}{P_i} = \frac{1}{s}\|A\|_F^2$$

Hence, we have $\mathbb{E}\|V - X\|_F^2 \leq \frac{k}{s}\|A\|_F^2$.

Let $F = TDX$, where X consists of rows $X^{(t)}$ defined above. Then, we have:

$$\begin{aligned}
\|A - F\|_{W,F} &\leq \left\|A - \hat{A}_k\right\|_{W,F} + \left\|\hat{A}_k - F\right\|_{W,F} \\
&\leq \left\|A - \hat{A}_k\right\|_{W,F} + \left\|\hat{A}_k - F\right\|_{F} \\
&= \left\|A - \hat{A}_k\right\|_{W,F} + \|DV - DX\|_F \\
&\leq \left\|A - \hat{A}_k\right\|_{W,F} + \|D\|_F \|V - X\|_F \\
&\leq \left\|A - \hat{A}_k\right\|_{W,F} + \left(\frac{k}{\kappa^2(A)w}\right)^{1/2} \|V - X\|_F
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Pr\left(\|A - F\|_{W,F} - \left\|A - \hat{A}_k\right\|_{W,F} \geq \frac{1}{4}\varepsilon \|A\|_{W,F}\right) \\
\leq &\Pr\left(\left(\frac{k}{\kappa^2(A)w}\right)^{1/2} \|V - X\|_F \geq \frac{1}{4}\varepsilon \|A\|_{W,F}\right) \\
= &\Pr\left(\frac{k}{\kappa^2(A)w} \|V - X\|_F^2 \geq \frac{1}{16}\varepsilon^2 \|A\|_{W,F}\right) \\
\leq &\Pr\left(\|V - X\|_F^2 \geq \frac{1}{16}\frac{\varepsilon^2 w^2 \kappa^2(A)}{k} \|A\|_F^2\right)
\end{aligned}$$

Setting $\frac{10k}{s} = \frac{1}{16}\frac{\varepsilon^2 w^2 \kappa^2(A)}{k}$, we get $s = O\left(\frac{k^2}{w^2\varepsilon^2\kappa^2(A)}\right)$ and

$$\Pr\left(\|V - X\|_F^2 \geq \frac{1}{16}\frac{\varepsilon^2 w^2 \kappa^2(A)}{k} \|A\|_F^2\right) \leq \frac{1}{100}$$

Therefore, with constant probability, we have

$$\|A - F\|_{W,F} < \left\|A - \hat{A}_k\right\|_{W,F} + \frac{1}{4}\varepsilon \|A\|_{W,F} \tag{2.1}$$

if we sample $O(\frac{k^2}{w^2\varepsilon^2\kappa^2(A)})$ rows.

Squaring both sides of Equation 2.1, we get:

$$\|A - F\|_{W,F}^2 < \left\|A - \hat{A}_k\right\|_{W,F}^2 + (\frac{1}{2}\varepsilon + \frac{1}{16}\varepsilon^2) \|A\|_{W,F}^2 \leq \left\|A - \hat{A}_k\right\|_{W,F}^2 + \varepsilon \|A\|_{W,F}^2$$

Since we can assume $\varepsilon < 1$[1]. □

[1]Otherwise, the 0 matrix is already a good approximation.

Now we have sampled the rows and built up matrix R where $R^{(t)} = A^{(i)}, i \in S, t = 1, \ldots, s$. We next want to solve a regression problem to get L such that matrix LR is the desired approximation.

Theorem 4. *There exists a regression algorithm that solves*

$$\arg\min_{L \in \mathbb{R}^{m \times s}} \|A - LR\|_{W,F}^2$$

Proof. We fold W into the norm, so we get:

$$\|A - LR\|_{W,F}^2 = \sum_j \sum_i (W_{ij}^{1/2} A_{ij} - \sum_k (W_{ij}^{1/2} R_{kj}) L_{ik})^2$$

Therefore, we can solve for each row of L separately. Now, solving for row i is just a regression problem $\arg\min_{x \in \mathbb{R}^s} \|A'x - b'\|$ where $A'_{jk} = W_{ij}^{1/2} R_{kj}$ and $b'_j = W_{ij}^{1/2} A_{ij}$. This can be done in $O(\text{nnz}(R) + \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon))$ time [4].

Since we need to solve m such problems, we need an estimate for $O(\text{nnz}(R) \cdot m)$. Notice that L contains sampled rows of A, so with constant probability, we have $O(\text{nnz}(R) \cdot m) = O(\text{nnz}(A) \cdot k')$ where k' is the number of rows we sample, so we get an algorithm of time complexity

$$O(\text{nnz}(A)k' + m \cdot \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon))$$

□

2.2.1 Algorithm

To build the algorithm, after we sampled the rows, we reduce the problem to a weighted regression problem, which we can solve easily. This will be as good as the matrix F analyzed above. In addition, this algorithm doesn't involve any knowledge of the optimum $\hat{A}_k$.

The time complexity of the algorithm is as follows: sampling has time complexity

$$O\left(\text{nnz}(A) + (m+n)\frac{k^2}{w^2\varepsilon^2\kappa^2(A)}\right)$$

And solving the regression problem has time complexity

$$O\left(\text{nnz}(A)\frac{k^2}{w^2\varepsilon^2\kappa^2(A)} + \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon)\right)$$

Hence, in total it takes $O(\text{nnz}(A)\frac{k^2}{w^2\varepsilon^2\kappa^2(A)} + \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon))$ time.

2.2.2 Further Improvement by leverage score sampling

In the last part of Theorem 4, we need to solve n regression problem. However, with leverage score sampling [8], we can sample the rows of $D_{W^{(i)1/2}}R^T$, where $D_{W^{(i)1/2}}$ is the diagonal matrix from row vector $W^{(i)}$, $i = 1, \ldots, m$.

Since the weights are in range $[w, 1]$, the leverage scores are the same for all i up to a factor of $w^{-1/2}$. In other words, we only need to oversample by a factor of $w^{-1/2}$ so that a single sampling and rescaling matrix $S \in \mathbb{R}^{s' \times m}$ with $s' = \tilde{O}(kw^{-3/2}\varepsilon^{-4})$ rows can be used to solve all regression problem. In brief, with constant probability, we can reach a $(1+\varepsilon)$ approximation by solving the new regression problem: $\arg\min_x \left\| SD_{W^{(i)1/2}}R^T x - SD_{W^{(i)1/2}}A^{(i)} \right\|^2$.

This algorithm for the second step has a time complexity of

$$O((\text{nnz}(L) + m \cdot \text{poly}(k, 1/w, 1/\varepsilon, 1/\kappa(A))) \log n)$$

for all m regression problems. Notice that matrix L has at most $n \cdot \text{poly}(k, 1/w, 1/\varepsilon, 1/\kappa(A))$ entries, so it takes at most $O((m+n) \cdot \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon))$ time.
Thus, in total the new algorithm has time complexity

$$O(\text{nnz}(A) + (m+n) \cdot \text{poly}(\log n, k, 1/w, 1/\kappa(A), 1/\varepsilon))$$

2.3 Refining to Multiplicative Error

We can further improve additive error to multiplicative error with two passes similar to [5].

Theorem 5. *Given W with entries in range $[w, 1]$ and A of full row rank, there exists an algorithm that finds matrix F with rank $O(\frac{k^2}{w^4\varepsilon^2\kappa^2(A)})$ s.t.*

$$\|A - F\|_{W,F}^2 \leq (1+\varepsilon) \left\| A - \hat{A}_k \right\|_{W,F}^2$$

where $\hat{A}_k$ is the optimum rank-k weighted approximation.

Proof. We first find a rank-k approximation B to A. Suppose $\|A - B\|_{W,F}^2 \leq c \left\| A - \hat{A}_k \right\|_{W,F}^2$. Note that a $O(1)$ unweighted approximation is a trivial $O(1/w)$ weighted approximation, so we can assume $c \in O(1/w)$.

This can be done by SVD to get an exact unweighted approximation of rank $k' = k$, or we can use the row/column selection algorithm in [3] to get a $(1+\varepsilon')$ approximation. Notice that this

algorithm requires selecting $O(k/\varepsilon')$ columns, but since we only need a $O(1)$ approximation, we can set $\varepsilon' = .5$ or any constant, so that we select only $k' = O(k)$ rows.

Then, we apply the algorithm in Theorem 3 with $k := k + k'$, $\varepsilon := \varepsilon w$ and $A := A - B$. This gives a matrix F' that satisfies:

$$\|A - B - F'\|_{W,F}^2 \leq \left\|A - B - \hat{X}\right\|_{W,F}^2 + \varepsilon w \|A - B\|_{W,F}^2$$

Where $\hat{X}$ denotes the optimal rank-$(k + k')$ approximation to $A - B$.

Notice that $\left\|A - B - \hat{X}\right\|_{W,F}^2 \leq \left\|A - \hat{A}_k\right\|_{W,F}^2$ because the latter takes minimum over a subset of matrices of the former.

Now, the matrix $F = B + F'$ is the matrix we want, with rank $O(\frac{k^2}{w^4\varepsilon^2\kappa^2(A)})$, and satisfy the relative error:

$$\|A - F\|_{W,F}^2 \leq (1 + \varepsilon) \left\|A - \hat{A}_k\right\|_{W,F}^2$$

The algorithm needs to sample $O(\frac{k^2}{w^4\varepsilon^2\kappa^2(A)})$ rows. □

2.3.1 Algorithm

The algorithm consists of two parts, finding a good rank-k approximation and running previous algorithm in Theorem 3.

The first part can be done with time complexity [3]

$$O(\text{nnz}(A) \log n + n \cdot \text{poly}(\log n, k, 1/\varepsilon))$$

While the second part has time complexity

$$O(\text{nnz}(A) + (m + n) \cdot \text{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A)))$$

Hence, in total it takes $O(\text{nnz}(A) \log n + (m + n) \cdot \text{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A)))$ time.

2.4 Lower Bound on the number of rows sampled

In this section we show that there exists a lower bound of the number of rows sampled if we use row sampling algorithm.

Theorem 6. *There exists examples where we need to sample at least $O(k/w)$ rows in order to get a good approximation.*

Proof. Set $m = n = \frac{k}{2w}$.
Let $A \in \mathbb{R}^{m \times n}$ be an identity matrix.

Separate the matrix $W \in \mathbb{R}^{m \times n}$ into $k \times k$ equally sized blocks. In other words, block i, j contains entries with row index in range $[\frac{i-1}{2w} + 1, \frac{i}{2w}]$ and column index in range $[\frac{j-1}{2w} + 1, \frac{j}{2w}]$ for $i, j = 1, \ldots, n$.
Then, we can set the entries of block i, j as follows:

$$W_{st}^{(i,j)} = \begin{cases} 1 & \\ w & \text{if } \quad i = j, s \neq t \\ 1 & \end{cases} \quad \begin{array}{l} i \neq j \\ \\ i = j, s = t \end{array}$$

In other words, for blocks on the diagonal, it contains 1 on its diagonal and w elsewhere. Otherwise, the block is filled with 1. The purpose is to concentrate the optimum into blocks on the diagonal.

By symmetry, the optimum must be of the form $U^T U$ where matrix U contains only 0 and 1. Then, we only need to decide how many 1's we need to have for each block. This is the reason why we set $n = \frac{k}{2w}$, because this implies each row of U contains all 1 for one block. Formally, we have:

$$U^{(i)} = \sum_{t=(i-1)/(2w)+1}^{i/(2w)} e_t$$

This gives an optimal cost of $\left(\frac{1}{4w^2} - \frac{1}{2w}\right) w = \frac{1}{4w} - \frac{1}{2}$ for one block.
Thus, in total we get an optimal cost of $\frac{k}{4w} - \frac{k}{2} \leq \frac{k}{4w}$.

Hence, for our algorithm, we start from an empty matrix, which is a 2-approximation, to build a $(1+\varepsilon)$-approximation. However, if we were to sample rows from A, each sample can only reveal one dimension. This implies no matter how we process the sampled rows, we at most cover s dimensions where s is the number of rows sampled. This gives a final cost of at least $\frac{k}{2w} - s$. Hence, we need to sample at least $O(\frac{k}{w})$ rows to reach $(\frac{k}{4w} - \frac{k}{2}) + \varepsilon \frac{k}{2w}$. □

2.5 Comparison with other additive error result

Algorithm 1 considers an approximation that doesn't have dependence on the weight matrix W. However, their conclusion depends on an assumed good approximation L. To reach optimum within additive error, it runs much slower than ours.

Proposition 7. *Consider the matrix in Section 2.4. Set $k = 1$ for simplification and let $L = uv^T$ where u, v are vectors.*
The algorithm in [2] must output a rank-$\Omega(\frac{k}{w\varepsilon^2})$ matrix to reach $\varepsilon \|A\|_{W,F}^2$ approximation.

Proof. First note that $\|A\|_{W,F} = \|A\|_F$ in this example.

Then, since their algorithm depends on the cost of a matrix L, we separate into two cases:

Case 1. If $\|A - L\|_{W,F}^2 \leq \left\|A - \hat{A}_k\right\|_{W,F}^2 + O(\varepsilon)\|A\|_{W,F}^2$, then by pigeon hole principle on the diagonal terms, it must be the case that $u_i v_i \in [1 - O(\varepsilon), 1 + O(\varepsilon)]$ for more than half of i's.

Therefore, consider the set $S = \{i \mid u_i v_i \in [1 - O(\varepsilon), 1 + O(\varepsilon)]\}$ which contains indices of such dimensions, we must have $\forall i, j \in S$, $u_i v_j \geq 1 - O(\varepsilon)$. This gives

$$\|L\|_F^2 = \Omega\left(\left(\frac{n}{2}\right)^2\right) = \Omega\left(\frac{1}{w^2}\right)$$

Hence, $\Lambda \geq \frac{\|L\|_F^2}{\|A\|_F^2} = \Omega(\frac{1}{w})$.

Moreover, notice that their algorithm chooses the principle component of the residual at each step, which contains one dimension only. Thus, their upper bound must be strict.

Hence, their theorem implies they need a rank-$\Theta(\frac{k\Lambda}{\varepsilon^2}) = \Omega(\frac{k}{w\varepsilon^2})$ matrix to reach a good additive error approximation.

Case 2. Otherwise, the algorithm cannot reach an additive error approximation with respect to the optimal, so we don't want to choose such a matrix L. □

Therefore, in extreme cases, our algorithm is worse than theirs by a factor of k/w since $\kappa(A) = 1$.

Finally, the time complexity of their algorithm can reach $O(\text{nnz}(A)kw^{-1}\varepsilon^{-2})$ in the example above. This is because they need to find the best direction at each step, which takes at least $O(\text{nnz}(A))$ time. Then, in extreme cases, they need to run $O(kw^{-1}\varepsilon^{-2})$ rounds to achieve additive error approximation.

However, we managed to improve dependence of $\text{nnz}(A)$ to $O(\text{nnz}(A))$ in the time complexity as described in Section 2.2.2, which is much better than theirs. In fact, this implies our algorithm is always better, given that the rank of the matrix can't be constant in almost all cases.

Chapter 3

Experiments

3.1 Data

We use synthetic data to test the algorithm. Specifically, we used normal distribution to generate the entries of matrix A. We also generate W as follows:

Let W'_{ij} be i.i.d. random variables following normal distribution, then

$$W_{ij} = .8\frac{{W'_{ij}}^2}{\max_{i',j'} {W'_{i'j'}}^2} + .2$$

So that we ensure matrix W has a minimum of $w \geq .2$.

3.2 Models

We compare the additive error algorithm as well as multiplicative error algorithm with the additive error approximation algorithm in [2].

Specifically, we follow the synthetic data generation procedure in Section 3.1 to generate matrix $W, A \in \mathbb{R}^{3000\times 3000}$.

Then, we test the algorithms to find approximations of rank from 5 to 20. In the second experiment, we allow 2 times the rank for our algorithms.

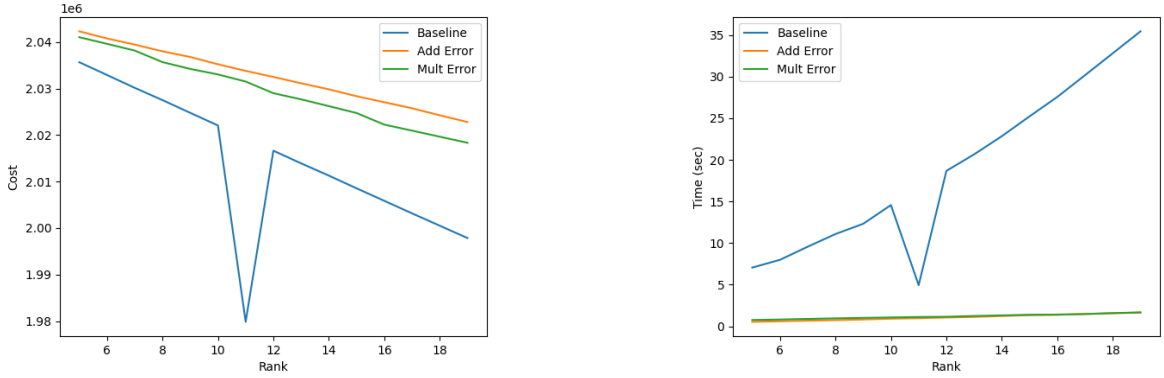
Moreover, notice that for multiplicative error approximation, we not only need the final rank, but also the rank for the first step. Hence, we set this rank to be $1/4$ of the final rank.

Finally, we didn't test leverage score sampling because it introduces more failure probability to the algorithm. In addition, for small cases, it is fast enough to solve the regression problems directly.

3.3 Experiment Results

3.3.1 Comparison with same rank

The first experiment gives following comparison on the cost function as well as runtime in Figure 3.1.



(a) Cost

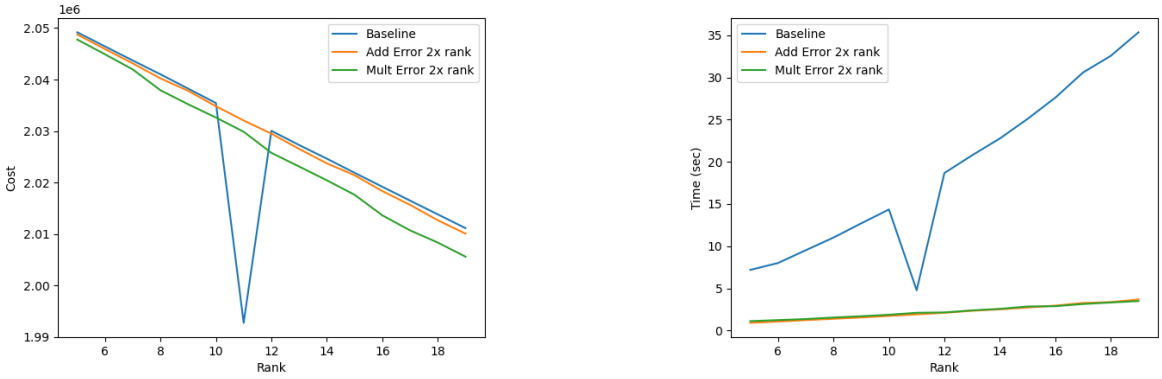
(b) Runtime

Figure 3.1: Comparison, same rank

This implies that although our algorithms didn't give better approximation, they are way faster than the algorithm in [2]. This implies that in most applications, we can output a matrix with more ranks in much less time, which motivates us to run the second experiment.

3.3.2 Comparison with 2x rank

The second experiment gives following comparison on the cost function as well as runtime in Figure 3.2.



(a) Cost

(b) Runtime

Figure 3.2: Comparison, 2x rank

This experiment implies that with 2 times the rank, we can do much better than baseline and still use much less time.

Chapter 4

Conclusion and Future Work

4.1 Weighted Low Rank Approximation

In conclusion, we showed there exists a row selection based algorithm runs in $O(\text{nnz}(A) + (m + n) \cdot \text{poly}(\log n, k, 1/w, 1/\varepsilon, 1/\kappa(A)))$ time that finds a weighted low rank approximation with the assumption that matrix A has full row rank and matrix W has a minimum positive entry. The number of rows sampled is $O(\frac{k^2}{w^2\varepsilon^2\kappa^2(A)})$ for additive error and $O(\frac{k^2}{w^4\varepsilon^2\kappa^2(A)})$ for multiplicative error.

These assumptions are, in some sense, necessary for the algorithm, because:

1. In the weighted settings, it is not necessary that the optimum is in the row span of the original matrix A. This implies it's possible that sampling algorithms may not find a good solution at all. Assuming A has full row rank could solve this problem.
2. If the matrix W has entry 0, then we cannot get a lower bound on the weighted Frobenius norm. Assuming a minimum positive weight can solve this problem.

As explained in introduction, there are applications with these assumptions, but we are also looking forward to find algorithms that don't need to make such assumptions.

4.2 Adding Regularizations

We are also seeking similar algorithms that can deal with variations that include regularizations terms. An example is Ridge regularization, defined as follows:

$$Cost(L) = \|A - L\|_{W,F}^2 + \lambda \|U\|_F^2 + \lambda \|V\|_F^2$$

Where $L = UV$, $U \in \mathbb{R}^{m \times \text{rank}(L)}$, $V \in \mathbb{R}^{\text{rank}(L) \times n}$.

With a simple argument, we can reduce this problem to:

$$Cost(L) = \|A - L\|_{W,F}^2 + 2\lambda \|L\|_1$$

Where $\|L\|_1$ is the Schatten-1 norm (trace norm).

In the unweighted case, the optimum can be found by subtracting λ from SVD result [11], while this won't work in weighted case. Therefore, we are looking forward to find efficient algorithms that can solve this problem in the future.

Bibliography

[1] Frank Ban, David Woodruff, and Richard Zhang. Regularized weighted low rank approximation. *Advances in neural information processing systems*, 32, 2019. 1

[2] Aditya Bhaskara, Aravinda Kanchana Ruwanpathirana, and Maheshakya Wijewardena. Additive error guarantees for weighted low rank approximation. In *International Conference on Machine Learning*, pages 874–883. PMLR, 2021. 1, 1, 1, 1, 9, 7, 3.2, 3.3.1

[3] Christos Boutsidis and David P Woodruff. Optimal cur matrix decompositions. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 353–362, 2014. 2.3, 2.3.1

[4] Kenneth L Clarkson and David P Woodruff. Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, 63(6):1–45, 2017. 1, 2.2

[5] Amit Deshpande, Luis Rademacher, Santosh S. Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2(12):225–247, 2006. doi: 10.4086/toc.2006.v002a012. URL `https://theoryofcomputing.org/articles/v002a012`. 2.3

[6] 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. 2

[7] Nicolas Gillis and François Glineur. Low-rank matrix approximation with weights or missing data is np-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011. 1

[8] Michael W. Mahoney. Randomized algorithms for matrices and data, 2011. 2.2.2

[9] Cameron Musco, Christopher Musco, and David P Woodruff. Simple heuristics yield provable algorithms for masked low-rank approximation. *arXiv preprint arXiv:1904.09841*, 2019. 1, 9

[10] Ilya Razenshteyn, Zhao Song, and David P Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 250–263, 2016. 1

[11] Madeleine Udell, Corinne Horn, Reza Zadeh, Stephen Boyd, et al. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016. 4.2

[12] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and*

Trends® in Theoretical Computer Science, 10(1–2):1–157, 2014. 1, 9

[13] Gale Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6(1): 49–53, 1941. 1