Tighter Low-rank Approximation via Sampling the Leveraged Element

Srinadh Bhojanapalli
The University of Texas at Austin
bsrinadh@utexas.edu

Prateek Jain
Microsoft Research, India
prajain@microsoft.com

Sujay Sanghavi
The University of Texas at Austin
sanghavi@mail.utexas.edu

Abstract

In this work, we propose a new randomized algorithm for computing a low-rank approximation to a given matrix. Taking an approach different from existing literature, our method first involves a specific biased sampling, with an element being chosen based on the leverage scores of its row and column, and then involves weighted alternating minimization over the factored form of the intended low-rank matrix, to minimize error only on these samples. Our method can leverage input sparsity, yet produce approximations in *spectral* (as opposed to the weaker Frobenius) norm; this combines the best aspects of otherwise disparate current results, but with a dependence on the condition number $\kappa = \sigma_1/\sigma_r$. In particular we require $O(nnz(M) + \frac{n\kappa^2 r^5}{\epsilon^2})$ computations to generate a rank-r approximation to M in spectral norm. In contrast, the best existing method requires $O(nnz(M) + \frac{nr^2}{\epsilon^4})$ time to compute an approximation in Frobenius norm. Besides the tightness in spectral norm, we have a better dependence on the error ϵ. Our method is naturally and highly parallelizable.

Our new approach enables two extensions that are interesting on their own. The first is a new method to directly compute a low-rank approximation (in efficient factored form) to the product of two given matrices; it computes a small random set of entries of the product, and then executes weighted alternating minimization (as before) on these. The sampling strategy is different because now we cannot access leverage scores of the product matrix (but instead have to work with input matrices). The second extension is an improved algorithm with smaller communication complexity for the distributed PCA setting (where each server has small set of rows of the matrix, and want to compute low rank approximation with small amount of communication with other servers).

1 Introduction

Finding a low-rank approximation to a matrix is fundamental to a wide array of machine learning techniques. The large sizes of modern data matrices has driven much recent work into efficient (typically randomized) methods to find low-rank approximations that do not exactly minimize the residual, but run much faster / parallel, with fewer passes over the data. Existing approaches involve either intelligent sampling of a few rows / columns of the matrix, projections onto lower-dimensional spaces, or sampling of entries followed by a top-r SVD of the resulting matrix (with unsampled entries set to 0).

We pursue a different approach: we first sample entries in a specific biased random way, and then minimize the error on these samples over a search space that is the factored form of the low-rank matrix we are trying to find. We note that this is different from approximating a 0-filled matrix; it is instead reminiscent of matrix completion in the sense that it only looks at errors on the sampled entries. Another crucial ingredient is how the sampling is done: we use a combination of ℓ_1 sampling, and of a distribution where the probability of an element is proportional to the sum of the leverage scores of its row and its column.

Both the sampling and the subsequent alternating minimization are naturally fast, parallelizable, and able to utilize sparsity in the input matrix. Existing literature has either focused on running in input sparsity time but approximation in (the weaker) Frobenius norm, or running in $O(n^2)$ time with approximation in spectral norm. Our method provides the best of both worlds: it runs in input sparsity time, with just two passes over the data matrix, and yields an approximation in spectral norm. It does however have a dependence on the ratio of the first to the rth singular value of the matrix.

Our alternative approach also yields new methods for two related problems: directly finding the low-rank approximation of the product of two given matrices, and distributed PCA.

Our contributions are thus three new methods in this space:

- **Low-rank approximation of a general matrix:** Our first (and main) contribution is a new method (LELA, Algorithm 1) for low-rank approximation of any given matrix: first draw a random subset of entries in a specific biased way, and then execute a weighted alternating minimization algorithm that minimizes the error on these samples

over a factored form of the intended low-rank matrix. The sampling is done with only two passes over the matrix (each in input sparsity time), and both the sampling and the alternating minimization steps are highly parallelizable and compactly stored/manipulated.

For a matrix M, let M_r be the best rank-r approximation (i.e. the matrix corresponding to top r components of SVD). Our algorithm finds a rank-r matrix $\widehat{M_r}$ in time $O(nnz(M) + \frac{n\kappa^2 r^5}{\epsilon^2})$, while providing approximation in spectral norm: $\|M - \widehat{M_r}\| \le \|M - M_r\| + \epsilon\|M - M_r\|_F$, where $\kappa = \sigma_1(M)/\sigma_r(M)$ is the condition number of M_r. Existing methods either can run in input sparsity time, but provide approximations in (the weaker) Frobenius norm (i.e. with $\|\cdot\|$ replaced by $\|\cdot\|_F$ in the above expression), or run in $O(n^2)$ time to approximate in spectral norm, but even then with leading constants larger than 1. Our method however does have a dependence on κ, which these do not. See Table 1 for a detailed comparison to existing results for low-rank approximation.

- **Direct approximation of a matrix product:** We provide a new method to directly find a low-rank approximation to the product of two matrices, without having to first compute the product itself. To do so, we first choose a small set of entries (in a biased random way) of the product that we will compute, and then again run weighted alternating minimization on these samples. The choice of the biased random distribution is now different from above, as the sampling step does not have access to the product matrix. However, again both the sampling and alternating minimization are highly parallelizable.

  For $A \in \mathbb{R}^{n_1 \times d}$, $B \in \mathbb{R}^{d \times n_2}$, and $n = max(n_1, n_2)$, our algorithm first chooses $O(nr^3 \log n/\epsilon^2)$ entries of the product $A \cdot B$ that it needs to sample; each sample takes $O(d)$ time individually, since it is a product of two length-d vectors (though these can be parallelized). The weighted alternating minimization then runs in $O(\frac{nr^5\kappa^2}{\epsilon^2})$ time (where $\kappa = \sigma_1(A \cdot B)/\sigma_r(A \cdot B)$). This results in a rank-r approximation $\widehat{AB_r}$ of $A \cdot B$ in spectral norm, as given above.

- **Distributed PCA:** Motivated by applications with really large matrices, recent work has looked at low-rank approximation in a distributed setting where there are s servers – each have small set of rows of the matrix – each of which can communicate with a central processor charged with coordinating

the algorithm. In this model, one is interested in find good approximations while minimizing both computations and the communication burden on the center.

We show that our LELA algorithm can be extended to the distributed setting while guaranteeing small communication complexity. In particular, our algorithm guarantees the same error bounds as that of our non-distributed version but guarantees communication complexity of $O(ds + \frac{nr^5\kappa^2}{\epsilon^2} \log n)$ real numbers for computing rank-r approximation to $M \in \mathbb{R}^{n \times d}$. For $n \approx d$ and large s, our analysis guarantees significantly lesser communication complexity than the state-of-the-art method [20], while providing tighter spectral norm bounds.

Notation: Capital letter M typically denotes a matrix. M^i denotes the i-th row of M, M_j denotes the j-th column of M, and M_{ij} denotes the (i,j)-th element of M. Unless specified otherwise, $M \in \mathbb{R}^{n \times d}$ and M_r is the best rank-r approximation of M. Also, $M_r = U^*\Sigma^*(V^*)^T$ denotes the SVD of M_r. $\kappa = \sigma_1^*/\sigma_r^*$ denotes the condition number of M_r, where σ_i^* is the i-th singular value of M. $\|u\|$ denotes the L_2 norm of vector u. $\|M\| = \max_{\|x\|=1}\|Mx\|$ denotes the spectral or operator norm of M. $\|M\|_F = \sqrt{\sum_{ij} M_{ij}^2}$ denotes the Frobenius norm of M. Also, $\|M\|_{1,1} = \sum_{ij} |M_{ij}|$. $dist(X, Y) = \|X_\perp^T Y\|$ denotes the principal angle based distance between subspaces spanned by X and Y orthonormal matrices. Typically, C denotes a global constant independent of problem parameters and can change from step to step.

Given a set $\Omega \subseteq [n] \times [d]$, $P_\Omega(M)$ is given by: $P_\Omega(M)(i, j) = M_{ij}$ if $(i, j) \in \Omega$ and 0 otherwise. $R_\Omega(M) = w. * P_\Omega(M)$ denotes the Hadamard product of w and $P_\Omega(M)$. That is, $R_\Omega(M)(i, j) = w_{ij}M_{ij}$ if $(i, j) \in \Omega$ and 0 otherwise. Similarly let $R_\Omega^{1/2}(M)(i, j) = \sqrt{w_{ij}}M_{ij}$ if $(i, j) \in \Omega$ and 0 otherwise.

2 Related results

Low rank approximation: Now we will briefly review some of the existing work on algorithms for low rank approximation. [12] introduced the problem of computing low rank approximation of a matrix M with few passes over M. They presented an algorithm that samples few rows and columns and does SVD to compute low rank approximation, and gave additive error guarantees. [8, 9] have extended these results. [1] considered a different approach based on entrywise sampling and quantization for low rank approximation and has given additive error bounds.

[16, 28, 10, 7] have given low rank approximation

Reference	Frobenius norm	Spectral norm	Computation time
BJS14 (Our Algorithm)	$(1+\epsilon)\|\Delta\|_F$	$\|\Delta\| + \epsilon\|\Delta\|_F$	$O(nnz(M) + \frac{nr^5\kappa^2\log(n)}{\epsilon^2})$
CW13[6]	$(1+\epsilon)\|\Delta\|_F$	$(1+\epsilon)\|\Delta\|_F$	$O(nnz(M) + \frac{nr^2}{\epsilon^4} + \frac{r^3}{\epsilon^5})$
BG13 [2]	$(1+\epsilon)\|\Delta\|_F$	$c\|\Delta\| + \epsilon\|\Delta\|_F$	$O(n^2(\frac{r+\log(n)}{\epsilon^2}) + n\frac{r^2\log(n)^2}{\epsilon^4})$
NDT09[26]	$(1+\epsilon)\|\Delta\|_F$	$c\|\Delta\| + \epsilon\sqrt{n}\|\Delta\|$	$O(n^2\log(\frac{r\log(n)}{\epsilon}) + \frac{nr^2\log(n)^2}{\epsilon^4})$
WLRT08[30]	$(1+\epsilon)\|\Delta\|_F$	$\|\Delta\| + \epsilon\sqrt{n}\|\Delta\|$	$O(n^2\log(\frac{r}{\epsilon}) + \frac{nr^4}{\epsilon^4})$
Sar06[28]	$(1+\epsilon)\|\Delta\|_F$	$(1+\epsilon)\|\Delta\|_F$	$O(nnz(M)\frac{r}{\epsilon} + n\frac{r^2}{\epsilon^2})$

Table 1: Comparison of error rates and computation time of some low rank approximation algorithms. $\Delta = M - M_r$.

algorithms with relative error guarantees in Frobenius norm. [30, 26] have provided guarantees on error in spectral norm which are later improved in [15, 2]. The main techniques of these algorithms is to use a random Gaussian or Hadamard transform matrix for projecting the matrix onto a low dimensional subspace and compute the rank-r subspace. [2] have given an algorithm based on random Hadamard transform that computes rank-r approximation in time $O(\frac{n^2r}{\epsilon^2})$ and gives spectral norm bound of $\|M - \widehat{M_r}\| \leq c\|M - M_r\| + \epsilon\|M - M_r\|_F$.

One drawback of Hadamard transform is that it cannot take advantage of sparsity of the input matrix. Recently [6] gave an algorithm using sparse subspace embedding that runs in input sparsity time with relative Frobenius norm error guarantees.

We presented some results in this area as a comparison with our results in table 1. This is a heavily subsampled set of existing results on low rank approximations. There is a lot of interesting work on very related problems of computing column/row based(CUR) decompositions, matrix sketching, low rank approximation with streaming data. Look at [25, 15] for more detailed discussion and comparison.

Matrix completion: Matrix completion problem is to recover a $n \times n$ rank-r matrix from observing small number of $(O(nr\log(n)))$ random entries. Nuclear norm minimization is shown to recover the matrix from uniform random samples if the matrix is incoherent [3, 4, 27, 14] . Similar results are shown for other algorithms like OptSpace [21] and alternating minimization [19, 17, 18]. Recently [5] has given guarantees for recovery of any matrix under leverage score sampling from $O(nr\log^2(n))$ entries.

Distributed PCA: In distributed PCA, one wants to compute rank-r approximation of a $n \times d$ matrix that is stored across s servers with small communication between servers. One popular model is row partition

model where subset of rows are stored at each server. Algorithms in [11, 23, 13, 22] achieve $O(\frac{dsr}{\epsilon})$ communication complexity with relative error guarantees in Frobenius norm, under this model.

Recently [20] have considered the scenario of arbitrary splitting of a $n \times d$ matrix and given an algorithm that has $O(\frac{dsr}{\epsilon})$ communication complexity with relative error guarantees in Frobenius norm.

3 Low-rank Approximation of Matrices

In this section we will present our main contribution: a new randomized algorithm for computing low-rank approximation of any given matrix. Our algorithm first samples a few elements from the given matrix $M \in \mathbb{R}^{n \times d}$, and then rank-$r$ approximation is computed using only those samples. Algorithm 1 provides a detailed pseudo-code of our algorithm; we now comment on each of the two stages:

Sampling: A crucial ingredient of our approach is using the correct sampling distribution. Recent results in matrix completion [5] indicate that a small number $(O(nr\log^2(n)))$ of samples drawn in a way biased by leverage scores[1] can capture *all* the information in any exactly low-rank matrix. While this is indicative, here we have neither access to the leverage scores, nor is our matrix exactly low-rank. We approximate the leverage scores with the row and column norms ($\|M^i\|^2$ and $\|M_j\|^2$), and account for the arbitrary high-rank nature of input by including an L_1 term in the sampling; the distribution is given in eq. (3.2). Computationally, our sampling procedure can be done in two passes and $O(m\log n + nnz(M))$ time.

Weighted alternating minimization: In our second step, we directly optimize over the factored form of the intended low-rank matrix, by minimizing a *weighted*

---

[1]If SVD of $M_r = U^*\Sigma^*(V^*)^T$ then leverage scores of M_r are $\|(U^*)^i\|^2$ and $\|(V^*)^j\|^2$ for all i,j.

squared error over the sampled elements from stage 1. That is, we first express the low-rank approximation $\widehat{M}_r$ as UV^T and then iterate over U and V alternatingly to minimize the weighted L_2 error over the *sampled* entries (see Sub-routine 2). Note that this is different from taking principal components of a 0-filled version of the sampled matrix. The weights give higher emphasis to elements with smaller sampling probabilities. In particular, the goal is to minimize the following objective function:

$$(3.1) \qquad Err(\widehat{M}_r) = \sum_{(i,j)\in\Omega} w_{ij} \left(M_{ij} - (\widehat{M}_r)_{ij} \right)^2,$$

where $w_{ij} = 1/\hat{q}_{ij}$ when $\hat{q}_{ij} > 0$, 0 else. For initialization of the WAltMin procedure, we compute SVD of $R_\Omega(M)$ (reweighed sampled matrix) followed by a trimming step (see Step 4, 5 of Sub-routine 2). Trimming step sets $(\widehat{U}^0)^i = 0$ if $\|(\widehat{U}^0)^i\| \geq 4\|M^i\|/\|M\|_F$ and $(\widehat{U}^0)^i = (U^0)^i$ otherwise; and prevents heavy rows/columns from having undue influence.

We now provide our main result for low-rank approximation and show that Algorithm 1 can provide a tight approximation to M_r while using a small number of samples $m = \mathbb{E}[|\Omega|]$.

THEOREM 3.1. *Let $M \in \mathbb{R}^{n\times d}$ be any given matrix ($n \geq d$) and let M_r be the best rank-r approximation to M. Set the number of samples $m = \frac{C}{\gamma}\frac{nr^3}{\epsilon^2}\kappa^2 \log(n)\log^2(\frac{\|M\|}{\zeta})$, where $C > 0$ is any global constant, $\kappa = \sigma_1/\sigma_r$ where σ_i is the i-th singular value of M. Also, set the number of iterations of WAltMin procedure to be $T = \log(\frac{\|M\|}{\zeta})$. Then, with probability greater than $1 - \gamma$ for any constant $\gamma > 0$, the output $\widehat{M}_r$ of Algorithm 1 with the above specified parameters m, T, satisfies:*

$$\|M - \widehat{M}_r\| \leq \|M - M_r\| + \epsilon \|M - M_r\|_F + \zeta.$$

That is, if $T = \log(\frac{\|M\|}{\epsilon\|M-M_r\|_F})$, we have:

$$\|M - \widehat{M}_r\| \leq \|M - M_r\| + 2\epsilon \|M - M_r\|_F.$$

Note that our time and sample complexity depends quadratically on κ. Recent results in the matrix completion literature shows that such a dependence can be improved to $\log(\kappa)$ by using a slightly more involved analysis [18]. We believe a similar analysis can be combined with our techniques to obtain tighter bounds; we leave a similar tighter analysis for future research as such a proof would be significantly more tedious and would take away from the key message of this paper.

3.1 Computation complexity: In the first step we take 2 passes over the matrix to compute the

---

Algorithm 1 LELA: Leveraged Element Low-rank Approximation

input matrix: $M \in \mathbb{R}^{n\times d}$, rank: r, number of samples: m, number of iterations: T
1: Sample $\Omega \subseteq [n] \times [d]$ where each element is sampled independently with probability: $\hat{q}_{ij} = \min\{q_{ij}, 1\}$

$$(3.2) \quad q_{ij} = m \cdot \left(\frac{\|M^i\|^2 + \|M_j\|^2}{2(n+d)\|M\|_F^2} + \frac{|M_{ij}|}{2\|M\|_{1,1}} \right).$$

/*See Section 3.1 for details about efficient implementation of this step*/
2: Obtain $P_\Omega(M)$ using one pass over M
3: $\widehat{M}_r = \mathsf{WAltMin}(P_\Omega(M), \Omega, r, \hat{q}, T)$
output $\widehat{M}_r$

Sub-routine 2 WAltMin: Weighted Alternating Minimization

input $P_\Omega(M)$, Ω, r, $\hat{q}$, T
1: $w_{ij} = 1/\hat{q}_{ij}$ when $\hat{q}_{ij} > 0$, 0 else, $\forall i, j$
2: Divide Ω in $2T+1$ equal uniformly random subsets, i.e., $\Omega = \{\Omega_0, \ldots, \Omega_{2T}\}$
3: $R_{\Omega_0}(M) \leftarrow w.* P_{\Omega_0}(M)$
4: $U^{(0)}\Sigma^{(0)}(V^{(0)})^T = SVD(R_{\Omega_0}(M), r)$ //Best rank-r approximation of $R_{\Omega_0}(M)$
5: Trim $U^{(0)}$ and let $\widehat{U}^{(0)}$ be the output (see Section 3)
6: **for** $t = 0$ to $T - 1$ **do**
7: $\quad \widehat{V}^{(t+1)} = \mathrm{argmin}_V \|R_{\Omega_{2t+1}}^{1/2}(M - \widehat{U}^{(t)}V^T)\|_F^2$, for $V \in \mathbb{R}^{d\times r}$.
8: $\quad \widehat{U}^{(t+1)} = \mathrm{argmin}_U \|R_{\Omega_{2t+2}}^{1/2}(M - U(\widehat{V}^{(t+1)})^T)\|_F^2$, for $U \in \mathbb{R}^{n\times r}$.
9: **end for**
output Completed matrix $\widehat{M}_r = \widehat{U}^{(T)}(\widehat{V}^{(T)})^T$.

---

sampling distribution (3.2) and sampling the entries based on this distribution. It is easy to show that this step would require $O(nnz(M) + m\log(n))$ time. Next, the initialization step of WAltMin procedure requires computing rank-r SVD of $R_{\Omega_0}(M)$ which has at most m non-zero entries. Hence, the procedure can be completed in $O(mr)$ time using standard techniques like power method. Note that by Lemma 3.1 we need top-r singular vectors of $R_{\Omega_0}(M)$ only upto constant approximation. Further t-th iteration of alternating minimization takes $O(2|\Omega_{2t+1}|r^2)$ time. So, the total time complexity of our method is $O(nnz(M) + mr^2)$. As shown in Theorem 3.1, our method requires $m = O(\frac{nr^3}{\epsilon^2}\kappa^2 \log(n)\log^2(\frac{\|M\|}{\epsilon\|M-M_r\|_F}))$ samples. Hence, the total run-time of our algorithm is: $O(nnz(M) + \frac{nr^5}{\epsilon^2}\kappa^2 \log(n)\log^2(\frac{\|M\|}{\epsilon\|M-M_r\|_F}))$.

Remarks: Now we will discuss how to sample entries of M using sampling method (3.2) in $O(nnz(M) + m\log(n))$ time. Consider the following multinomial based sampling model: sample the number of elements per row (say m_i) by doing m draws using a multinomial distribution over the rows, given by $\{0.5(\frac{d\|M^i\|^2}{(n+d)\|M\|_F^2} + \frac{1}{n+d}) + 0.5\frac{\|M^i\|_1}{\|M\|_{1,1}}\}$. Then, sample m_i elements of the row-i, using $\{0.5\frac{\|M_j\|^2}{\|M\|_F^2} + 0.5\frac{|M_{ij}|}{\|M\|_{1,1}}\}$ over $j \in [d]$, with replacement.

The failure probability in this model is bounded by 2 times the failure probability if the elements are sampled according to (3.2) [3]. Hence, we can instead use the above mentioned multinomial model for sampling. Moreover, $\|M^i\|$, $\|M^i\|_1$ and $\|M_j\|$ can be computed in time $O(nnz(M)+n)$, so m_i's can be sampled efficiently. Moreover, the multinomial distribution for all the rows can be computed in time $O(d + nnz(M))$, $O(d)$ work for setting up the first $\|M_j\|$ term and $nnz(M)$ term for changing the base distribution wherever M_{ij} is non-zero. Hence, the total time complexity is $O(m+n+nnz(M))$.

3.2 Proof Overview:
We now present the key steps in our proof of Theorem 3.1. As mentioned in the previous section, our algorithm proceeds in two steps: entry-wise sampling of the given matrix M and then weighted alternating minimization (WAltMin) to obtain a low-rank approximation of M.

Hence, the goal is to analyze the WAltMin procedure, with input samples obtained using (3.2), to obtain the bounds in Theorem 3.1. Now, WAltMin is an iterative procedure solving an inherently non-convex problem, $\min_{U,V} \sum_{(i,j)\in\Omega} w_{ij}(\mathbf{e}_i^T UV^T \mathbf{e}_j - M_{ij})^2$. Hence, it is prone to local minimas or worse, might not even converge. However, recent results for low-rank matrix completion have shown that alternating minimization (with appropriate initialization) can indeed be analyzed to obtain exact matrix completion guarantees.

Our proof also follows along similar lines, where we show that the initialization procedure (step 4 of Sub-routine 2) provides an accurate enough estimate of M_r and then at each step, we show a geometric decrease in distance to M_r. However, our proof differs from the previous works in two key aspects: a) existing proof techniques of alternating minimization assume that each element is sampled uniformly at random, while we can allow biased and approximate sampling, b) existing techniques crucially use the assumption that M_r is incoherent, while our proof avoids this assumption using the weighted version of AltMin.

We now present our bounds for initialization as well as for each step of the WAltMin procedure. Theorem 3.1 follows easily from the two bounds.

LEMMA 3.1. (INITIALIZATION) *Let the set of entries Ω be generated according to (3.2). Also, let $m \geq C\frac{n}{\delta^2}\log(n)$. Then, the following holds (w.p. $\geq 1 - \frac{2}{n^{10}}$):*

$$(3.3) \qquad \|R_\Omega(M) - M\| \leq \delta\|M\|_F.$$

Also, if $\|M-M_r\|_F \leq \frac{1}{576\kappa r^{1.5}}\|M_r\|_F$, then the following holds (w.p. $\geq 1 - \frac{2}{n^{10}}$):

$$\|(\widehat{U}^{(0)})^i\| \leq 8\sqrt{r}\sqrt{\|M^i\|^2/\|M\|_F^2} \quad and$$
$$dist(\widehat{U}^{(0)}, U^*) \leq \frac{1}{2},$$

where $\widehat{U}^{(0)}$ is the initial iterate obtained using Steps 4, 5 of Sub-Procedure 2. $\kappa = \sigma_1^*/\sigma_r^*$, σ_i^* is the i-th singular value of M, $M_r = U^*\Sigma^*(V^*)^T$.

Let $\mathcal{P}_r(A)$ be the best rank-r approximation of A. Then, Lemma 3.1 and Weyl's inequality implies that:

$$(3.4) \quad \|M - \mathcal{P}_r(R_\Omega(M))\| \leq \|M - R_\Omega(M)\|$$
$$+ \|R_\Omega(M) - \mathcal{P}_r(R_\Omega(M))\| \leq \|M - M_r\| + 2\delta\|M\|_F.$$

Now, we can have two cases: 1) $\|M - M_r\|_F \geq \frac{1}{576\kappa r^{1.5}}\|M_r\|_F$: In this case, setting $\delta = \epsilon/(\kappa r^{1.5})$ in (3.4) already implies the required error bounds of Theorem 3.1[2]. 2) $\|M-M_r\|_F \leq \frac{1}{576\kappa r^{1.5}}\|M_r\|_F$. In this regime, we will show now that alternating minimization reduces the error from initial $\delta\|M\|_F$ to $\delta\|M - M_r\|_F$.

LEMMA 3.2. (WALTMIN DESCENT) *Let hypotheses of Theorem 3.1 hold. Also, let $\|M - M_r\|_F \leq \frac{1}{576\kappa r\sqrt{r}}\|M_r\|_F$. Let $\widehat{U}^{(t)}$ be the t-th step iterate of Sub-Procedure 2 (called from Algorithm 1), and let $\widehat{V}^{(t+1)}$ be the $(t + 1)$-th iterate (for V). Also, let $\|(U^{(t)})^i\| \leq 8\sqrt{r}\kappa\sqrt{\frac{\|M_j\|^2}{\|M\|_F^2} + \frac{|M_{ij}|}{\|M\|_F}}$ and $dist(\widehat{U}^{(t)}, U^*) \leq \frac{1}{2}$, where $U^{(t)}$ is a set of orthonormal vectors spanning $\widehat{U}^{(t)}$. Then, the following holds (w.p. $\geq 1 - \gamma/T$):*

$$dist(\widehat{V}^{(t+1)}, V^*) \leq \frac{1}{2}dist(\widehat{U}^{(t)}, U^*) + \epsilon\|M - M_r\|_F/\sigma_r^*,$$

and $\|(V^{(t+1)})^j\| \leq 8\sqrt{r}\kappa\sqrt{\frac{\|M_j\|^2}{\|M\|_F^2} + \frac{|M_{ij}|}{\|M\|_F}}$, where $V^{(t+1)}$ is a set of orthonormal vectors spanning $\widehat{V}^{(t+1)}$.

The above lemma shows that distance between $\widehat{V}^{(t+1)}$ and V^* (and similarly, $\widehat{U}^{(t+1)}$ and U^*) decreases geometrically up to $\epsilon\|M - M_r\|_F/\sigma_r^*$. Hence, after

---

[2]There is a small technicality here: alternating minimization can potentially worsen this bound. But the error after each step of alternating minimization can be effectively checked using a small cross-validation set and we can stop if the error increases.

$\log(\|M\|_F/\zeta)$ steps, the first error term in the bounds above vanishes and the error bound given in Theorem 3.1 is obtained.

Note that, the sampling distribution used for our result is a "hybrid" distribution combining leverage scores and the L_1-style sampling. However, if M is indeed a rank-r matrix, then our analysis can be extended to handle the leverage score based sampling itself ($q_{ij} = m \cdot \frac{\|M^i\|^2 + \|M_j\|^2}{2n\|M\|_F^2}$). Hence our results also show that weighted alternating minimization can be used to solve the coherent-matrix completion problem introduced in [5].

3.3 Direct Low-rank Approximation of Matrix Product

In this section we present a new algorithm for the following problem: suppose we are given two matrices, and desire a low-rank approximation of their product AB; in particular, we are *not* interested in the actual full matrix product itself (as this may be unwieldy to store and use, and thus wasteful to produce in its entirety). One example setting where this arises is when one wants to calculate the joint counts between two very large sets of entities; for example, web companies routinely come across settings where they need to understand (for example) how many users both searched for a particular query and clicked on a particular advertisement. The number of possible queries and ads is huge, and finding this co-occurrence matrix from user logs involves multiplying two matrices – query-by-user and user-by-ad respectively – each of which is itself large.

We give a method that directly produces a low-rank approximation of the final product, and involves storage and manipulation of only the efficient factored form (i.e. one tall and one fat matrix) of the final intended low-rank matrix. Note that as opposed to the previous section, the matrix does not already exist and hence we do not have access to its row and column norms; so we need a new sampling scheme (and a different proof of correctness).

Algorithm: Suppose we are given an $n_1 \times d$ matrix A and another $d \times n_2$ matrix B, and we wish to calculate a rank-r approximation of the product $A \cdot B$. Our algorithm proceeds in two stages:

1. Choose a biased random set $\Omega \subset [n_1] \times [n_2]$ of elements as follows: choose an intended number m (according to Theorem 3.2 below) of sampled elements, and then independently include each $(i,j) \in [n_1] \times [n_2]$ in Ω with probability given by $\hat{q}_{ij} = \min\{1, q_{ij}\}$ where

$$(3.5) \qquad q_{ij} := m \cdot \left(\frac{\|A^i\|^2}{n_2\|A\|_F^2} + \frac{\|B_j\|^2}{n_1\|B\|_F^2} \right),$$

Then, find $P_\Omega(A \cdot B)$, i.e. only the elements of the product AB that are in this set Ω.

2. Run the alternating minimization procedure WAltMin($P_\Omega(A \cdot B), \Omega, r, \hat{q}, T$), where T is the number of iterations (again chosen according to Theorem 3.2 below). This produces the low-rank approximation in factored form.

Remarks: Note that the sampling distribution now depends only on the row norms $\|A^i\|^2$ of A and the column norms $\|B_j\|^2$ of B; each of these can be found completely in parallel, with one pass over each row/column of the matrices A / B. A second pass, again parallelizable, calculates the element $(A \cdot B)_{ij}$ of the product, for $(i,j) \in \Omega$. Once this is done, we are again in the setting of doing weighted alternating minimization over a small set of samples – the setting we had before, and as already mentioned this too is highly parallelizable and very fast overall. In particular, the computation complexity of the algorithm is $O(|\Omega| \cdot (d + r^2)) = O(m(d + r^2)) = O(\frac{nr^3\kappa^2}{\epsilon^2} \cdot (d + r^2))$ (suppressing terms dependent on norms of A and B), where $n = \max\{n_1, n_2\}$.

We now present our theorem on the number of samples and iterations needed to make this procedure work with at least a constant probability.

THEOREM 3.2. *Consider matrices $A \in \mathbb{R}^{n_1 \times d}$ and $B \in \mathbb{R}^{d \times n_2}$ and let $m = \frac{C}{\gamma} \cdot \frac{(\|A\|_F^2 + \|B\|_F^2)^2}{\|AB\|_F^2} \cdot \frac{nr^3}{(\epsilon)^2}\kappa^2 \log(n) \log^2(\frac{\|A\|_F + \|B\|_F}{\zeta})$, where $\kappa = \sigma_1^*/\sigma_r^*$, σ_i^* is the i-th singular value of $A \cdot B$ and $T = \log(\frac{\|A\|_F + \|B\|_F}{\zeta})$. Let Ω be sampled using probability distribution (3.5). Then, the output $\widehat{AB}_r = WAltMin(P_\Omega(A \cdot B), \Omega, r, \hat{q}, T)$ of Sub-routine 2 satisfies (w.p. $\geq 1 - \gamma$): $\|A \cdot B - \widehat{AB}_r\| \leq \|A \cdot B - (A \cdot B)_r\| + \epsilon\|A \cdot B - (A \cdot B)_r\|_F + \zeta$.*

Next, we show an application of our matrix-multiplication approach to approximating covariance matrices $M = YY^T$, where Y is a $n \times d$ sample matrix. Note, that a rank-r approximation to M can be computed by computing low rank approximation of Y, $\widehat{Y}_r$, i.e., $\tilde{M}_r = \widehat{Y}_r\widehat{Y}_r^T$. However, as we show below, such an approach leads to weaker bounds as compared to computing low rank approximation of YY^T:

COROLLARY 3.1. *Let $M = YY^T \in \mathbb{R}^{n \times n}$ and let Ω be sampled using probability distribution (3.5) with $m \geq \frac{C}{\gamma}\frac{nr^3}{\epsilon^2}\kappa^2 \log(n) \log^2(\frac{\|Y\|}{\zeta})$, the output $\widehat{M}_r$ of $WAltMin(P_\Omega(M), \Omega, r, \hat{q}, \log(\frac{\|Y\|}{\zeta}))$ satisfy (w.p. $\geq 1 - \gamma$):*

$$\|\widehat{M}_r - (YY^T)_r\| \leq \epsilon \|Y - Y_r\|_F^2 + \zeta.$$

Further when $\|Y - Y_r\|_F \leq \|Y_r\|_F$ we get:

$$\|\widehat{M}_r - (YY^T)_r\| \leq \epsilon \left\|YY^T - (YY^T)_r\right\|_F + \zeta.$$

Now, one can compute $\tilde{M}_r = \widehat{Y}_r \widehat{Y}_r^T$ in time $O(n^2 r + \frac{nr^5}{\epsilon^2})$ with error $\frac{\|(YY^T)_r - \tilde{M}_r\|}{\|Y\|^2} \leq \epsilon \frac{\|Y - Y_r\|_F}{\|Y\|}$. Where as computing low rank approximation of YY^T gives(from Corollary 3.1) $\frac{\|(YY^T)_r - \widehat{M}_r\|}{\|Y\|^2} \leq \epsilon \frac{\|YY^T - Y_r Y_r^T\|_F}{\|Y\|^2}$, which can be much smaller than $\epsilon \frac{\|Y - Y_r\|_F}{\|Y\|}$. The difference in error is a consequence of larger gap between singular values of YY^T compared to Y. For more related applications see section 4.5 of [15].

4 Distributed Principal Component Analysis

Modern large-scale systems have to routinely compute PCA of data matrices with millions of data points embedded in similarly large number of dimensions. Now, even storing such matrices on a single machine is not possible and hence most industrial scale systems use distributed computing environment to handle such problems. However, performance of such systems depend not only on computation and storage complexity, but also on the required amount of communication between different servers.

In particular, we consider the following distributed PCA setting: Let $M \in \mathbb{R}^{n \times d}$ be a given matrix (assume $n \geq d$ but $n \approx d$). Also, let M be row partitioned among s servers and let $M_{\mathbf{r_k}} \in \mathbb{R}^{n \times d}$ be the matrix with rows $\{r_k\} \subseteq [n]$ of M, stored on k-th server. Moreover, we assume that one of the servers act as Central Processor(CP) and in each round all servers communicate with the CP and the CP communicates back with all the servers. Now, the goal is to compute $\widehat{M}_r$, an estimate of M_r, such that the total communication (i.e. number of bits transferred) between CP and other servers is minimized. Note that, such a model is now standard for this problem and was most recently studied by [20].

Recently several interesting results [11, 13, 22, 20] have given algorithms to compute rank-r approximation of M, $\tilde{M}_r$ in the above mentioned distributed setting. In particular, [20] proposed a method that for row-partitioned model requires $O(\frac{dsr}{\epsilon} + \frac{sr^2}{\epsilon^4})$ communication to obtain a relative Frobenius norm guarantee,

$$||M - \tilde{M}_r||_F \leq (1 + \epsilon)||M - M_r||_F.$$

In contrast, a distributed setting extension of our LELA algorithm 1 has linear communication complexity $O(ds + \frac{nr^5}{\epsilon^2})$ and computes rank-r approximation $\widehat{M}_r$, with $||M - \widehat{M}_r|| \leq ||M - M_r|| + \epsilon||M - M_r||_F$. Now note that if $n \approx d$ and if s scales with n (which

is a typical requirement), then our communication complexity can be significantly better than that of [20]. Moreover, our method provides spectral norm bounds as compared to relatively weak Frobenius bounds mentioned above.

Algorithm: The distributed version of our LELA algorithm depends crucially on the following observation: given V, each row of U can be updated independently. Hence, servers need to communicate rows of V only. There also, we can use the fact that each server requires only $O(nr/s \log n)$ rows of V to update their corresponding $U_{\mathbf{r_k}}$. $U_{\mathbf{r_k}}$ denote restriction of U to rows in set $\{r_k\}$ and 0 outside and similarly $\widehat{V}_{\mathbf{c_k}}, \widehat{Y}_{\mathbf{c_k}}$ denote restriction of $V, \widehat{Y}$ to rows $\{c_k\}$.

We now describe the distributed version of each of the critical step of LELA algorithm. See Algorithm 3 for a detailed pseudo-code. For simplicity, we dropped the use of different set of samples in each iteration. Correspondingly the algorithm will modify to distributing samples Ω^k into $2T + 1$ buckets and using one in each iteration. This simplification doesn't change the communication complexity.

Sampling: For sampling, we first compute column norms $\|M_j\|, \forall 1 \leq j \leq d$ and communicate to each server. This operation would require $O(ds)$ communication. Next, each server (server k) samples elements from its rows $\{\mathbf{r_k}\}$ and stores $R_{\Omega^k}(M_{\mathbf{r_k}})$ locally. Note that because of *independence over rows*, the servers don't need to transmit their samples to other servers.

Initialization: In the initialization step, our algorithm computes top r right singular vector of $R_\Omega(M)$ by iterations $\widehat{Y}^{(t+1)} = R_\Omega(M)^T R_\Omega(M) Y^t = \sum_k R_{\Omega^k}(M)^T R_{\Omega^k}(M) Y^t$. Now, note that computing $R_{\Omega^k}(M) Y^t$ requires server k to access atmost $|\Omega^k|$ columns of Y^t. Hence, the total communication from the CP to all the servers in this round is $O(|\Omega|r)$. Similarly, each column of $R_{\Omega^k}(M)^T R_{\Omega^k}(M) Y^t$ is only $|\Omega^k|$ sparse. Hence, total communication from all the servers to CP in this round is $O(|\Omega|r)$. Now, we need constant many rounds to get a constant factor approximation to SVD of $R_\Omega(M)$, which is enough for good initialization in WAltMin procedure. Hence, total communication complexity of the initialization step would be $O(|\Omega|r)$.

Alternating Minimization Step: For alternating minimization, update to rows of U is computed at the corresponding servers and the update to V is computed at the CP. For updating $\widehat{U}_{\mathbf{r_k}}^{(t+1)}$ at server k, we use the following observation: updating $\widehat{U}_{\mathbf{r_k}}^{(t+1)}$ requires atmost $|\Omega^k|$ rows of $\widehat{V}^{(t)}$. Hence, the total communication from CP to all the servers in the t-th iteration is $O(|\Omega|r)$. Next, we make a critical observation that update $\widehat{V}^{(t+1)}$

can be computed by adding certain messages from each server (see Algorithm 3 for more details). Message from server k to CP is of size $O(|\Omega^k|r^2)$. Hence, total communication complexity in each round is $O(|\Omega|r^2)$ and total number of rounds is $O(\log(\|M\|_F/\zeta))$.

We now combine the above given observations to provide error bounds and communication complexity of our distributed PCA algorithm:

THEOREM 4.1. *Let the $n \times d$ matrix M be distributed over s servers according to the row-partition model. Let $m \geq \frac{C}{\gamma} \frac{nr^3}{\epsilon^2} \kappa^2 \log(n) \log^2(\frac{\|M_r\|}{\zeta})$. Then, the algorithm 3 on completion will leave matrices $\widehat{U}_{\mathbf{r_k}}^{(t+1)}$ at server k and $\widehat{V}^{(t+1)}$ at CP such that the following holds (w.p. $\geq 1-\gamma$): $\|M-\widehat{U}^{(t+1)}(\widehat{V}^{(t+1)})^T\| \leq \|M-M_r\|+\epsilon\|M-M_r\|_F + \zeta$, where $\widehat{U}^{(t+1)} = \sum_k \widehat{U}_{\mathbf{r_k}}^{(t+1)}$. This algorithm has a communication complexity of $O(ds + |\Omega|r^2) = O(ds + \frac{nr^5\kappa^2}{\epsilon^2} \log^2(\frac{\|M_r\|}{\zeta}))$ real numbers.*

As discussed above, each update to $\widehat{V}^{(t)}$ and $\widehat{U}^{(t)}$ are computed exactly as given in the WAltMin procedure (Sub-routine 2). Hence, error bounds for the algorithm follows directly from Theorem 3.1. Communication complexity bounds follows by observing that $|\Omega| \leq 2m$ w.h.p.

Remark: The sampling step given above suggests another simple algorithm where we can compute $P_\Omega(M)$ in a distributed fashion and communicate the samples to CP. All the computation is performed at CP afterwards. Hence, the total communication complexity would be $O(ds + |\Omega|) = O(ds + \frac{nr^3\kappa^2}{\epsilon^2} \log(\frac{\|M_r\|}{\zeta}))$, which is lesser than the communication complexity of Algorithm 3. However, such an algorithm is not desirable in practice, because it is completely reliant on one single server to perform all the computation. Hence it is slower and is fault-prone. In contrast, our algorithm can be implemented in a peer-peer scenario as well and is more fault-tolerant.

Also, the communication complexity bound of Theorem 4.1 only bounds the total number of real numbers transferred. However, if each of the number requires several bits to communicate then the real communication can still be very large. Below, we bound each of the real number that we transfer, hence providing a bound on the number of bits transferred.

Bit complexity: First we will bound $w_{ij}M_{ij}$. Note that we need to bound this only for $(i,j) \in \Omega$. Now, $|w_{ij}M_{ij}| \leq \|R_\Omega(M)\|_\infty \leq \|R_\Omega(M)\| \leq \|M\| + \epsilon\|M\|_F \leq 2*ndM_{max}$, where the third inequality follows from Lemma 3.1. Hence if the entries of the matrix M are being represented using b bits initially then the

algorithm needs to use $O(b+\log(nd))$ bits. By the same argument we get a bound of $O(b+\log(nd))$ bits for computing $\|M^i\|^2, \forall i; \|M_j\|^2, \forall j; \|M\|_F^2$ and $\|M\|_{1,1}$.

Further at any stage of the WAltMin iterations $\|\widehat{U}^{(t)}(\widehat{V}^{(t+1)})^T\|_\infty \leq \|\widehat{U}^{(t)}(\widehat{V}^{(t+1)})^T\| \leq 2\|M\|_F$. So this stage also needs $O(b+\log(n))$ bits for computation. Hence overall the bit complexity of each of the real numbers of the algorithm 3 is $O(b + \log(nd))$, if b bits are needed for representing the matrix entries. That is, overall communication complexity of the algorithm is $O((b + \log(nd)) \cdot (ds + \frac{nr^3\kappa^2}{\epsilon^2} \log(\frac{\|M_r\|}{\zeta}))$.

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001.

[2] C. Boutsidis and A. Gittens. Improved matrix algorithms via the subsampled randomized hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1301–1340, 2013.

[3] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[4] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[5] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In *Proceedings of The 31st International Conference on Machine Learning*, pages 674–682, 2014.

[6] K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 81–90. ACM, 2013.

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

[8] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *SODA*, volume 3, pages 223–232, 2003.

[9] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[10] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 316–326. Springer, 2006.

[11] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1434–1453. SIAM, 2013.

[12] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 370–378, Nov 1998.

[13] M. Ghashami and J. M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *SODA*, pages 707–717. SIAM, 2014.

[14] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *Information Theory, IEEE Transactions on*, 57(3):1548–1566, 2011.

[15] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[16] S. Har-Peled. Low rank matrix approximation in linear time. *Manuscript. http://valis. cs. uiuc. edu/sariel/papers/05/lrank/lrank. pdf*, 2006.

[17] M. Hardt. Understanding alternating minimization for matrix completion. *arXiv preprint arXiv:1312.0925*, 2013.

[18] M. Hardt and M. Wootters. Fast matrix completion without the condition number. In *Proceedings of The 27th Conference on Learning Theory*, pages 638–678, 2014.

[19] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 665–674. ACM, 2013.

[20] R. Kannan, S. S. Vempala, and D. P. Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of The 27th Conference on Learning Theory*, pages 1040–1057, 2014.

[21] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

[22] Y. Liang, M.-F. Balcan, and V. Kanchanapally. Distributed pca and k-means clustering. In *The Big Learning Workshop at NIPS*, 2013.

[23] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588. ACM, 2013.

[24] L. Mackey, M. I. Jordan, R. Y. Chen, B. Farrell, and J. A. Tropp. Matrix concentration inequalities via the method of exchangeable pairs. *arXiv preprint arXiv:1201.6002*, 2012.

[25] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

[26] N. H. Nguyen, T. T. Do, and T. D. Tran. A fast and efficient algorithm for low-rank approximation of a matrix. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 215–224. ACM, 2009.

[27] B. Recht. A simpler approach to matrix completion. *arXiv preprint arXiv:0910.0651*, 2009.

[28] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 143–152. IEEE, 2006.

[29] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[30] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.

A Concentration Inequalities

In this section we will review couple of concentration inequalities we use in the proofs.

LEMMA A.1. (BERNSTEIN'S INEQUALITY) *Let $X_1, ... X_n$ be independent scalar random variables. Let $|X_i| \leq L, \forall i$ w.p. 1. Then,*

$$(A.1) \quad \mathbb{P}\left[\left|\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} \mathbb{E}[X_i]\right| \geq t\right]$$
$$\leq 2\exp\left(\frac{-t^2/2}{\sum_{i=1}^{n}\text{Var}(X_i) + Lt/3}\right).$$

LEMMA A.2. (MATRIX BERNSTEIN'S INEQ. [29]) *Let $X_1, ... X_p$ be independent random matrices in $\mathbb{R}^{n \times n}$. Assume each matrix has bounded deviation from its mean:*

$$\|X_i - \mathbb{E}[X_i]\| \leq L, \forall i \text{ w.p. } 1.$$

Also let the variance be

$$\sigma^2 = \max\left\{\left\|\mathbb{E}\left[\sum_{i=1}^{p}(X_i - \mathbb{E}[X_i])(X_i - \mathbb{E}[X_i])^T\right]\right\|,\right.$$
$$\left.\left\|\mathbb{E}\left[\sum_{i=1}^{p}(X_i - \mathbb{E}[X_i])^T(X_i - \mathbb{E}[X_i])\right]\right\|\right\}.$$

Then,
$$(A.2)$$
$$\mathbb{P}\left[\left\|\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i])\right\| \geq t\right] \leq 2n\exp\left(\frac{-t^2/2}{\sigma^2 + Lt/3}\right).$$

Recall the Shatten-p norm of a matrix X is

$$\|X\|_p = \left(\sum_{i=1}^{n}\sigma_i(X)^p\right)^{1/p}.$$

$\sigma_i(X)$ is the ith singular value of X. In particular for $p = 2$, Shatten-2 norm is the Frobenius norm of the matrix.

LEMMA A.3. *[Matrix Chebyshev Inequality [24]] Let X be a random matrix. For all $t > 0$,*

$$(A.3) \qquad \mathbb{P}\left[\|X\| \geq t\right] \leq \inf_{p \geq 1} t^{-p} \mathbb{E}\left[\|X\|_p^p\right].$$

B   Proofs of section 3

In this section we will present proof for Theorem 3.1. For simplicity we will only discuss proofs for the case when matrix is square. Rectangular case is a simple extension. We will provide proofs of the supporting lemmas first.

We will recall some of the notation now.

$$q_{ij} = m \cdot \left(0.5 \frac{\|M^i\|^2 + \|M_j\|^2}{2n\|M\|_F^2} + 0.5 \frac{|M_{ij}|}{\|M\|_{1,1}} \right).$$

Let $\hat{q}_{ij} = \min\{q_{ij}, 1\}$. This is to make sure the probabilities are all less than 1. Recall the definition of weights $w_{ij} = 1/\hat{q}_{ij}$ when $\hat{q}_{ij} > 0$ and 0 else. Note that $\sum_{ij} \hat{q}_{ij} \leq m$. Also let $m \geq \beta nr \log(n)$.

Let $\{\delta_{ij}\}$ be the indicator random variables and $\delta_{ij} = 1$ with probability $\hat{q}_{ij}$. Define Ω to be the sampling operator with $\Omega_{ij} = \delta_{ij}$. Define the weighted sampling operator R_Ω such that, $R_\Omega(M)_{ij} = \delta_{ij} w_{ij} M_{ij}$.

Throughout the proofs we will drop the subscript of Ω that denotes different sampling sets at each iteration of WAltMin.

First we will abstract out the properties of the sampling distribution (3.2) that we use in the rest of the proof.

LEMMA B.1. *For Ω generated according to (3.2) and under the assumptions of Lemma 3.1 the following holds, for all (i, j) such that $q_{ij} \leq 1$.*

$$(B.4) \qquad \frac{M_{ij}}{\hat{q}_{ij}} \leq \frac{2n}{m}\|M\|_F,$$

$$(B.5) \qquad \sum_{\{j:\hat{q}_{ij}=q_{ij}\}} \frac{M_{ij}^2}{\hat{q}_{ij}} \leq \frac{4n}{m}\|M\|_F^2,$$

$$(B.6) \qquad \frac{\|(U^*)^i\|^2}{\hat{q}_{ij}} \leq \frac{8nr\kappa^2}{m},$$

and

$$(B.7) \qquad \frac{\|(U^*)^i\|\|(V^*)^j\|}{\hat{q}_{ij}} \leq \frac{8nr\kappa^2}{m}.$$

The proof of the lemma B.1 is straightforward from the definition of q_{ij}.

B.1   Initialization: Now we will provide proof of the initialization lemma 3.1.
Proof of lemma 3.1:

Proof. The proof of this lemma has two parts.
1) We show that

$$\|R_\Omega(M) - M\| \leq \delta \|M\|_F$$

2) We show that the trimming step of algorithm 2 gives the required row norm bounds on $\widehat{U}^{(0)}$.

$$\|(\widehat{U}^{(0)})^i\| \leq 8\sqrt{r}\sqrt{\|M^i\|^2/\|M\|_F^2} \quad \text{and}$$

$$dist(\widehat{U}^{(0)}, U^*) \leq \frac{1}{2},$$

Proof of the first step: We prove the proof of the first part using the matrix Bernstein inequality. Note that the $L1$ term in the sampling distribution will help in getting good bounds on absolute magnitude of random variables X_{ij} in this proof.

Let $X_{ij} = (\delta_{ij} - \hat{q}_{ij}) w_{ij} M_{ij} e_i e_j^T$. Note that $\{X_{ij}\}_{i,j=1}^n$ are independent zero mean random matrices. Also $R_\Omega(M) - \mathbb{E}\left[R_\Omega(M)\right] = \sum_{ij} X_{ij}$.

First we will bound $\|X_{ij}\|$. When $q_{ij} \geq 1$, $\hat{q}_{ij} = 1$ and $\delta_{ij} = 1$, and $X_{ij} = 0$ with probability 1. Hence we only need to consider cases when $\hat{q}_{ij} = q_{ij} \leq 1$. We will assume this in all the proofs without explicitly mentioning it any more.

$$(B.8) \quad \|X_{ij}\| = \max\{|(1 - \hat{q}_{ij}) w_{ij} M_{ij}|, |\hat{q}_{ij} w_{ij} M_{ij}|\}.$$

Recall $w_{ij} = 1/\hat{q}_{ij}$. Hence

$$|(1 - \hat{q}_{ij}) w_{ij} M_{ij}| \leq \left| \frac{M_{ij}}{\hat{q}_{ij}} \right| \overset{\zeta_1}{\leq} \frac{2n}{m}\|M\|_F.$$

ζ_1 follows from (B.4).

$$|\hat{q}_{ij} w_{ij} M_{ij}| = |M_{ij}| \overset{\zeta_1}{\leq} \left| \frac{M_{ij}}{\hat{q}_{ij}} \right| \leq \frac{2n}{m}\|M\|_F.$$

ζ_1 follows from $\hat{q}_{ij} \leq 1$.

Hence, $\|X_{ij}\|$ is bounded by $L = \frac{2n}{m}\|M\|_F$. Now we will bound the variance.

$$\left\| \mathbb{E}\left[\sum_{ij} X_{ij} X_{ij}^T \right] \right\| = \left\| \mathbb{E}\left[\sum_{ij} (\delta_{ij} - \hat{q}_{ij})^2 w_{ij}^2 M_{ij}^2 e_i e_i^T \right] \right\|$$

$$= \left\| \sum_{ij} \hat{q}_{ij}(1 - \hat{q}_{ij}) w_{ij}^2 M_{ij}^2 e_i e_i^T \right\|$$

$$= \max_i \left| \sum_j \hat{q}_{ij}(1 - \hat{q}_{ij}) w_{ij}^2 M_{ij}^2 \right|.$$

Now,

$$\sum_j \hat{q}_{ij}(1-\hat{q}_{ij})w_{ij}^2 M_{ij}^2 \le \sum_j \frac{M_{ij}^2}{(\hat{q}_{ij})} \overset{\zeta_1}{\le} \frac{4n}{m}\|M\|_F^2.$$

ζ_1 follows from (B.5). Hence

$$\left\|\mathbb{E}\left[\sum_{ij} X_{ij}X_{ij}^T\right]\right\| = \max_i \left|\sum_j \hat{q}_{ij}(1-\hat{q}_{ij})w_{ij}^2 M_{ij}^2\right|$$
$$\le \max_i \frac{4n}{m}\|M\|_F^2 = \frac{4n}{m}\|M\|_F^2.$$

We can prove the same bound for the $\left\|\mathbb{E}\left[\sum_{ij} X_{ij}^T X_{ij}\right]\right\|$.
Hence $\sigma^2 = \frac{4n}{m}\|M\|_F^2$. Now using matrix Bernstein inequality with $t = \delta\|M\|_F$ gives, with probability $\ge 1 - \frac{2}{n^2}$,

$$\|R_\Omega(M) - \mathbb{E}[R_\Omega(M)]\| = \|R_\Omega(M) - M\| \le \delta\|M\|_F.$$

Hence we get $\|M - \mathcal{P}_r(R_\Omega(M))\| \le \|M - R_\Omega(M)\| + \|R_\Omega(M) - \mathcal{P}_r(R_\Omega(M))\| \le \|M - M_r\| + 2\delta\|M\|_F$, which implies $\|M_r - \mathcal{P}_r(R_\Omega(M))\| \le 2\|M - M_r\| + 2\delta\|M\|_F$.
Let SVD of $\mathcal{P}_r(R_\Omega(M))$ be $U^{(0)}\Sigma^{(0)}(V^{(0)})^T$. Hence,

$$\|\mathcal{P}_r(R_\Omega(M)) - M_r\|^2 = \|U^{(0)}\Sigma^{(0)}(V^{(0)})^T - U^*\Sigma^*(V^*)^T\|^2$$
$$\ge \|(I - U^{(0)}(U^{(0)})^T)U^*\Sigma^*(V^*)^T\|^2 \ge (\sigma_r^*)^2\|(U_\perp^{(0)})^T U^*\|^2$$

This implies $dist(U^{(0)}, U^*) \le \frac{2\|M - M_r\| + 2\delta\|M\|_F}{\sigma_r^*} \le \frac{1}{144r}$. This follows from the assumption $\|M - M_r\|_F \le \frac{1}{576\kappa r^{1.5}}\|M_r\|_F$ and $\delta \le \frac{1}{576\kappa r^{1.5}}$. $\kappa = \frac{\sigma_1^*}{\sigma_r^*}$ is the the condition number of M_r.

Proof of the trimming step: From previous step we know that $\|R_\Omega(M) - M\| \le \delta\|M\|_F$ and consequently $dist(U^{(0)}, U^*) \le \delta_2$. Let,

$$l_i = \sqrt{4\|M^i\|^2/\|M\|_F^2},$$

be the estimates for the left leverages scores of the matrix M. Since $\|M - M_r\|_F \le \|M_r\|_F$, $l_i^2 \ge \frac{\sum_{k=1}^r (\sigma_k^*)^2(U_{ik}^*)^2}{\sum_{k=1}^r (\sigma_k^*)^2}$.

Set the elements bigger than $2l_i$ in the ith row of U^0 to 0 and let $\tilde{U}$ be the new initialization matrix obtained. Also since $dist(U^{(0)}, U^*) \le \delta_2$, for every $j = 1, .., r$ there exists a vector $\bar{u}_j$ in $\mathbb{R}^n$, such that $\langle U_j^{(0)}, \bar{u}_j\rangle \ge \sqrt{1 - \delta_2^2}$, $\|\bar{u}_j\| = 1$ and $|(\bar{u}_j)_i|^2 \le \frac{\sum_{k=1}^r (\sigma_k^*)^2(U_{ik}^*)^2}{\sum_{k=1}^r (\sigma_k^*)^2}$ for all i. Now $\tilde{U}_j$ is the jth column of $\tilde{U}$ obtained by setting the entries of the jth column of $U^{(0)}$ to 0 whenever the ith entry of $U_j^{(0)}$ is bigger than $2l_i$. For such i,

$$(B.9) \quad \left|(\tilde{U}_j)_i - (\bar{u}_j)_i\right| \le \sqrt{\frac{\sum_{k=1}^r (\sigma_k^*)^2(U_{ik}^*)^2}{\sum_{k=1}^r (\sigma_k^*)^2}}$$
$$\le \left|(U_j^0)_i - (\bar{u}_j)_i\right|,$$

since $\left|(U_j^{(0)})_i - (\bar{u}_j)_i\right| \ge 2l_i - l_i = \sqrt{\frac{\sum_{k=1}^r (\sigma_k^*)^2(U_{ik}^*)^2}{\sum_{k=1}^r (\sigma_k^*)^2}}$.
For the rest of the coordinates, $(\tilde{U}_j)_i = (U_j^{(0)})_i$. Hence,

$$\left\|\tilde{U}_j - \bar{u}_j\right\| \le \left\|U_j^{(0)} - \bar{u}_j\right\| = \sqrt{2 - 2\langle U_j^{(0)}, \bar{u}_j\rangle} \le \sqrt{2}\delta_2.$$

Hence $\left\|\tilde{U}_j\right\| \ge 1 - \sqrt{2}\delta_2$, and $\left\|U_j^{(0)} - \tilde{U}_j\right\| \le \sqrt{1 - \left\|\tilde{U}_j\right\|^2} \le 2\sqrt{\delta_2}$, for $\delta_2 \le \frac{1}{\sqrt{2}}$. Also $\left\|U^{(0)} - \tilde{U}\right\|_F \le 2\sqrt{r\delta_2}$. This gives abound on the smallest singular value of $\tilde{U}$.

$$\sigma_{\min}(\tilde{U}) \ge \sigma_{\min}(U^{(0)}) - \sigma_{\max}(\tilde{U} - U^{(0)}) \ge 1 - 2\sqrt{r\delta_2}.$$

Now let the reduced QR decomposition of $\tilde{U}$ be $\tilde{U} = \widehat{U}^{(0)}\Lambda^{-1}$, where $\widehat{U}^{(0)}$ is the matrix with orthonormal columns. From the bounds above we get that

$$\|\Lambda\|^2 = \frac{1}{\sigma_{\min}(\widehat{U}^{(0)}\Lambda^{-1})^2} = \frac{1}{\sigma_{\min}(\tilde{U})^2} \le 4,$$

when $\delta_2 \le \frac{1}{16r}$.
First we will show that this trimming step will not increase the distance to U^* by much. To bound the $dist(\widehat{U}^{(0)}, U^*)$ consider, $\|(u_\perp^*)^T U\|$, where $u_\perp^*$ is some vector perpendicular to U^*.

$$\|(u_\perp^*)^T \widehat{U}^{(0)}\| = \|(u_\perp^*)^T \tilde{U}\Lambda\|$$
$$\le (\|(u_\perp^*)^T U^{(0)}\| + \|(u_\perp^*)^T (\tilde{U} - U^{(0)})\|)\|\Lambda\|$$
$$\le (\delta_2 + 2\sqrt{r\delta_2})2 \le 6\sqrt{r\delta_2} \le \frac{1}{2},$$

for $\delta_2 \le \frac{1}{144r}$. Second we will bound $\|(\widehat{U}^{(0)})^i\|$.

$$\|(\widehat{U}^{(0)})^i\| = \|e_i^T \widehat{U}^{(0)}\| = \|e_i^T \tilde{U}\Lambda\|$$
$$\le \|e_i^T \tilde{U}\|\|\Lambda\| \le 2l_i\sqrt{r}2 \le 8\sqrt{r}\sqrt{\|M^i\|^2/\|M\|_F^2}.$$

Hence we finish the proof of the second part of the lemma.

B.2 Weighted AltMin analysis: We first provide proof of Lemma 3.2 for rank-1 case to explain the main ideas and in next section we will discuss rank-r case. Hence $M_1 = \sigma^* u^*(v^*)^T$. Before the proof of the lemma we will prove couple of supporting lemmas.

Let u^t and v^{t+1} be the normalized vectors of the iterates $\widehat{u}^t$ and $\widehat{v}^{t+1}$ of WAltMin. We assume that samples for each iteration are generated independently. For simplicity we will drop the subscripts on Ω that denote different set of samples in each iteration in the rest of the proof.

The weighted alternating minimization updates at the $t + 1$ iteration are,
(B.10)
$$\|\widehat{u}^t\|\widehat{v}_j^{t+1} = \sigma^* v_j^* \frac{\sum_i \delta_{ij} w_{ij} u_i^t u_i^*}{\sum_i \delta_{ij} w_{ij} (u_i^t)^2} + \frac{\sum_i \delta_{ij} w_{ij} u_i^t (M - M_1)_{ij}}{\sum_i \delta_{ij} w_{ij} (u_i^t)^2}.$$

Writing in terms of power method updates we get,
(B.11)
$$\|\widehat{u}^t\|\widehat{v}^{t+1} = \sigma^* \langle u^*, u^t\rangle v^* - \sigma^* B^{-1}(\langle u^t, u^*\rangle B - C)v^* + B^{-1}y,$$

where B and C are diagonal matrices with $B_{jj} = \sum_i \delta_{ij} w_{ij} (u_i^t)^2$ and $C_{jj} = \sum_i \delta_{ij} w_{ij} u_i^t u_i^*$ and y is the vector $R_\Omega(M - M_1)^T u^t$ with entries $y_j = \sum_i \delta_{ij} w_{ij} u_i^t (M - M_1)_{ij}$.

Now we will bound the error caused by the $M - M_1$ component in each iteration.

LEMMA B.2. *For Ω generated according to (3.2) and under the assumptions of Lemma 3.2 the following holds:*
(B.12)
$$\left\|(u^t)^T R_\Omega(M - M_1) - (u^t)^T(M - M_1)\right\| \leq \delta\|M - M_1\|_F,$$

with probability greater that $1 - \frac{\gamma}{T\log(n)}$, for $m \geq \beta n\log(n)$, $\beta \geq \frac{4c_1^2 T}{\gamma\delta^2}$. Hence,$\left\|(u^t)^T R_\Omega(M - M_1)\right\| \leq dist(u^t, u^*)\|M - M_1\| + \delta\|M - M_1\|_F$, for constant δ.

Proof. Let the random matrices $X_{ij} = (\delta_{ij} - \hat{q}_{ij})w_{ij}(M - M_1)_{ij}(u^t)_i e_j^T$. Then $\sum_{ij} X_{ij} = (u^t)^T R_\Omega(M - M_1) - (u^t)^T(M - M_1)$. Also $\mathbb{E}[X_{ij}] = 0$. We will use the matrix Chebyshev inequality for $p = 2$. Now we will bound $\mathbb{E}\left[\left\|\sum_{ij} X_{ij}\right\|_2^2\right]$.

$$\mathbb{E}\left[\left\|\sum_{ij} X_{ij}\right\|_2^2\right]$$
$$= \mathbb{E}\left[\sum_j \left(\sum_i (\delta_{ij} - \hat{q}_{ij})w_{ij}(M - M_1)_{ij}(u^t)_i\right)^2\right]$$
$$\overset{\zeta_1}{=} \sum_j \sum_i \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij})^2(M - M_1)_{ij}^2(u_i^t)^2$$
$$\leq \sum_{ij} w_{ij}(u_i^t)^2(M - M_1)_{ij}^2 \overset{\zeta_2}{\leq} \frac{4nc_1^2}{m}\|M - M_1\|_F^2.$$

ζ_1 follows from the fact that X_{ij} are zero mean independent random variables. ζ_2 follows from (B.15). Hence applying the matrix Chebyshev inequality for $p = 2$ and $t = \delta\|M - M_1\|_F$ gives the result.

LEMMA B.3. *For Ω sampled according to (3.2) and under the assumptions of Lemma 3.2 the following*

holds:
(B.13)
$$\left|\sum_j \delta_{ij} w_{ij}(u_j^*)^2 - \sum_j (u_j^*)^2\right| \leq \delta_1,$$

with probability greater that $1 - \frac{2}{n^2}$, for $m \geq \beta n\log(n)$, $\beta \geq \frac{16}{\delta_1^2}$ and $\delta_1 \leq 3$.

Proof. Note that $\left|\sum_j \delta_{ij} w_{ij}(u_j^*)^2 - \sum_j (u_j^*)^2\right| = \left|\sum_j (\delta_{ij} - \hat{q}_{ij})w_{ij}(u_j^*)^2\right|$. Let the random variable $X_j = (\delta_{ij} - \hat{q}_{ij})w_{ij}(u_j^*)^2$. $\mathbb{E}[X_j] = 0$ and $Var(X_j) = \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij}(u_j^*)^2)^2$. Hence,

$$\sum_j Var(X_j) = \sum_j \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij}(u_j^*)^2)^2$$
$$= \sum_j (\frac{1}{\hat{q}_{ij}} - 1)(u_j^*)^4 \leq \sum_j \frac{(u_j^*)^2}{\hat{q}_{ij}}(u_j^*)^2$$
$$\overset{\zeta_1}{\leq} \frac{16n}{m}\sum_j (u_j^*)^2 = \frac{16n}{m}.$$

ζ_1 follows from (B.6). Also it is easy to check that $|X_j| \leq \frac{16n}{m}$. Now applying Bernstein inequality gives the result.

LEMMA B.4. *For Ω sampled according to (3.2) and under the assumptions of Lemma 3.2 the following holds:*
(B.14)
$$\|(\langle u^t, u^*\rangle B - C)v^*\| \leq \delta_1\sqrt{1 - \langle u^*, u^t\rangle^2},$$

with probability greater than $1 - \frac{2}{n^2}$, for $m \geq \beta n\log(n), \beta \geq \frac{48c_1^2}{\delta_1^2}$ and $\delta_1 \leq 3$.

Proof. Let
$$\alpha_i = u_i^t(\langle u^*, u^t\rangle u_i^t - u_i^*).$$
Hence the jth coordinate of the error term in equation (B.11) is
$$\frac{\sum_i \delta_{ij} w_{ij} \alpha_i v_j^*}{\sum_i \delta_{ij} w_{ij}(u_i^t)^2}.$$
Recall that $\alpha_i = u_i^t(\langle u^*, u^t\rangle u_i^t - u_i^*)$. Let $X_{ij} = \delta_{ij} w_{ij} \alpha_i v_j^* e_j e_1^T$, for i, j in $[1, ..n]$. Note that X_{ij} are independent random matrices. Then $(\langle u^t, u^*\rangle B - C)v^*$ is the first and the only column of the matrix $\sum_{i,j=1}^n X_{ij}$. We will bound $\|\sum_{ij=1}^n X_{ij}\|$ using matrix Bernstein inequality.

$$\sum_{ij} \mathbb{E}[X_{ij}] = \sum_j \sum_i \hat{q}_{ij} w_{ij} \alpha_i v_j^* e_j e_1^T$$
$$= \sum_j \sum_i \alpha_i v_j^* e_j e_1^T = 0,$$

because $\sum_i \alpha_i = 0$. Now we will give a bound on $\|X_{ij}\|$.

$$\|X_{ij}\| = |\delta_{ij} w_{ij} \alpha_i v_j^*| \leq \frac{v_j^* u_i^t}{\hat{q}_{ij}} (\langle u^*, u^t \rangle u_i^t - u_i^*)$$

$$\overset{\zeta_1}{\leq} \frac{16nc_1}{m} \sqrt{\sum_i (\langle u^*, u^t \rangle u_i^t - u_i^*)^2} = \frac{16nc_1}{m} \sqrt{1 - \langle u^*, u^t \rangle^2}$$

ζ_1 follows from (B.7) and (B.15).

Now we will bound the variance.

$$\left\| \mathbb{E} \left[\sum_{ij} (X_{ij} - \mathbb{E}[X_{ij}])^T (X_{ij} - \mathbb{E}[X_{ij}]) \right] \right\|$$

$$= \left\| \mathbb{E} \left[\sum_{ij} (X_{ij} T X_{ij} - \mathbb{E}[X_{ij}]^T \mathbb{E}[X_{ij}]) \right] \right\|$$

$$= \left\| \sum_j \sum_i \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij}\alpha_i v_j^*)^2 e_1 e_1^T \right\|.$$

Then,

$$\sum_i \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij}\alpha_i v_j^*)^2$$

$$\leq \sum_i w_{ij}(u_i^t)^2(\langle u^*, u^t \rangle u_i^t - u_i^*)^2 (v_j^*)^2$$

$$\leq \frac{c_1^2 4n}{m}(v_j^*)^2 \sum_i (\langle u^*, u^t \rangle u_i^t - u_i^*)^2$$

$$\leq \frac{c_1^2 4n}{m}(v_j^*)^2 (1 - \langle u^*, u^t \rangle^2).$$

Hence,

$$\left\| \mathbb{E} \left[\sum_{ij} (X_{ij} - \mathbb{E}[X_{ij}])^T (X_{ij} - \mathbb{E}[X_{ij}]) \right] \right\| \leq \frac{4nc_1^2}{m}(1 - \langle u^*, u^t \rangle^2)$$

The lemma follows from applying matrix Bernstein inequality.

Now we will provide proof of lemma 3.2.
Proof of lemma 3.2:[Rank-1 case]

Proof. Now we will prove that the distance between u^t, v^{t+1} and u^*, v^* decreases with each iteration. Recall from the assumptions of the Lemma we have the following row norm bounds for u^t;

(B.15) $|u_i^t| \leq c_1 \sqrt{\|M^i\|^2/\|M\|_F^2 + |M_{ij}|/\|M\|_F}.$

First we will prove that $dist(u^t, u^*)$ decreases in each iteration and second we will prove that v^{t+1} satisfies this condition.

Bounding $\langle v^{t+1}, v^* \rangle$:
Using Lemma B.3, Lemma B.4 and equation (B.11) we get,

(B.16) $\|\widehat{u}^t\| \langle \widehat{v}^{t+1}, v^* \rangle$

$$\geq \sigma^* \langle u^t, u^* \rangle - \sigma^* \frac{\delta_1}{1 - \delta_1} \sqrt{1 - \langle u^*, u^t \rangle^2} - \frac{1}{1 - \delta_1} \|y^T v^*\|$$

and

(B.17)
$$\|\widehat{u}^t\| \langle \widehat{v}^{t+1}, v_\perp^* \rangle \leq \sigma^* \frac{\delta_1}{1 - \delta_1} \sqrt{1 - \langle u^*, u^t \rangle^2} + \frac{1}{1 - \delta_1} \|y\|.$$

Hence by applying the noise bounds Lemma B.2 we get,

$$dist(v^{t+1}, v^*)^2 = 1 - \langle v^{t+1}, v^* \rangle^2$$

$$= \frac{\langle \widehat{v}^{t+1}, v_\perp^* \rangle^2}{\langle \widehat{v}^{t+1}, v_\perp^* \rangle^2 + \langle \widehat{v}^{t+1}, v^* \rangle^2} \leq \frac{\langle \widehat{v}^{t+1}, v_\perp^* \rangle^2}{\langle \widehat{v}^{t+1}, v^* \rangle^2}$$

$$\overset{\zeta_1}{\leq} 25(\delta_1 dist(u^t, u^*) + dist(u^t, u^*)\|M - M_1\|/\sigma^* + \delta\|M - M_1\|_F/\sigma^*)^2.$$

ζ_1 follows from $\delta_1 \leq \frac{1}{2}$, $\langle u^t, u^* \rangle \geq \langle u^0, u^* \rangle$ and $(\langle u^*, u^0 \rangle - 2\delta_1 \sqrt{1 - \langle u^*, u^0 \rangle^2} \geq \frac{1}{2}$, $\delta \leq \frac{1}{20}$ and $\delta_1 \leq \frac{1}{20}$. Hence,

$$dist(v^{t+1}, v^*) \leq \frac{1}{2}dist(u^t, u^*) + 5\delta\|M - M_1\|_F/\sigma^*.$$

Bounding $|v_j^{t+1}|$:
From Lemma B.3 and (B.15) we get that $\left| \sum_i \delta_{ij} w_{ij}(u_i^t)^2 - 1 \right| \leq \delta_1$ and $|\sum_i \delta_{ij} w_{ij} u_i^* u_i^t - \langle u^*, u^t \rangle| \leq \delta_1$, when $\beta \geq \frac{16c_1^2}{\delta_1^2}$. Hence,

(B.18) $1 - \delta_1 \leq B_{jj} = \sum_i \delta_{ij} w_{ij}(u_i^t)^2 \leq 1 + \delta_1,$

and

(B.19) $C_{jj} = \sum_i \delta_{ij} w_{ij} u_i^t u_i^* \leq \langle u^t, u^* \rangle + \delta_1.$

Recall that

$$\|\widehat{u}^t\| |\widehat{v}_j^{t+1}| = \left| \frac{\sum_i \delta_{ij} w_{ij} u_i^t M_{ij}}{\sum_i \delta_{ij} w_{ij}(u_i^t)^2} \right| \leq \frac{1}{1 - \delta_1} \sum_i \delta_{ij} w_{ij} u_i^t M_{ij}.$$

We will bound using $\sum_i \delta_{ij} w_{ij} u_i^t M_{ij}$ using Bernstein inequality.

Let $X_i = (\delta_{ij} - \hat{q}_{ij}) w_{ij} u_i^t M_{ij}$. Then $\sum_i \mathbb{E}[X_i] = 0$ and $\sum_i u_i^t M_{ij} \leq \|M_j\|$ by Cauchy-Schwartz inequality.

$\sum_i \mathrm{Var}(X_i) = \sum_i \hat{q}_{ij}(1 - \hat{q}_{ij})(w_{ij})^2(u_i^t)^2 M_{ij}^2 \le \sum_i w_{ij}(u_i^t)^2 M_{ij}^2 \le \frac{4nc_1^2}{m}\|M_j\|^2$. Finally $|X_{ij}| \le |w_{ij}u_i^t M_{ij}| \le \frac{4nc_1}{m}|M_{ij}|/\sqrt{\frac{|M_{ij}|}{\|M\|_F}} \le \frac{4nc_1}{m}\sqrt{|M_{ij}|\|M\|_F}$.

Hence applying Bernstein inequality with $t = \delta\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}$ gives, $\sum_i \delta_{ij}w_{ij}u_i^t M_{ij} \le (1+\delta_1)\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}$ with probability greater than $1 - \frac{2}{n^3}$ when $m \ge \frac{24c_1^2}{\delta_1^2} n\log(n)$.

Since $\delta_1 \le \frac{1}{20}$, we get, $\|\hat{u}^t\||\hat{v}_j^{t+1}| \le \frac{21}{19}\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}$. Now we will bound $\|\hat{v}^{t+1}\|$.

$$\|\hat{u}^t\|\|\hat{v}^{t+1}\| \ge \|\hat{u}^t\|\langle\hat{v}^{t+1}, v^*\rangle$$
$$\overset{\zeta_1}{\ge} \sigma^*\langle u^t, u^*\rangle - \sigma^*\frac{\delta_1}{1-\delta_1}\sqrt{1 - \langle u^*, u^t\rangle^2} - \frac{1}{1-\delta_1}\|y^T v^*\|$$
$$\overset{\zeta_2}{\ge} \sigma^*\langle\hat{u}^0, u^*\rangle - 2\sigma^*\delta_1\sqrt{1 - \langle u^*, \hat{u}^0\rangle^2} - 2\delta\|M - M_1\|$$
$$\overset{\zeta_3}{\ge} \frac{2}{5}\sigma^*.$$

ζ_1 follows from Lemma B.4 and equations (B.11) and (B.18). ζ_2 follows from using $\langle u^*, \hat{u}^0\rangle \le \langle u^*, u^t\rangle$ and $\delta_1 \le \frac{1}{20}$. ζ_3 follows from the argument: for $\delta_1 \le \frac{1}{16}$, $\langle\hat{u}^0, u^*\rangle - 2\delta_1\sqrt{1 - \langle u^*, \hat{u}^0\rangle^2}$ is greater than $\frac{1}{2}$, if $\langle\hat{u}^0, u^*\rangle \ge \frac{3}{5}$. This holds because $dist(u^*, \hat{u}^0) \le \frac{4}{5}$ from Lemma 3.1.

Hence we get

$$v_j^{t+1} = \frac{\hat{v}_j^{t+1}}{\|\hat{v}^{t+1}\|} \le 3\frac{\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}}{\sigma^*}$$
$$\le c_1\sqrt{\|M_j\|^2/\|M\|_F^2 + |M_{ij}|/\|M\|_F},$$

for $c_1 = 6$.

Hence we have shown that v^{t+1} satisfies the row norm bounds. From Lemma B.2 we have, in each iteration with probability greater than $1 - \frac{\gamma}{T\log(n)}$, $\|(u^t)^T R_\Omega(M - M_1)\| \le dist(u^t, u^*)\|M - M_1\| + \delta\|M - M_1\|_F$. Hence the probability of failure in T iterations is less than γ. Lemma now follows from assumption on m.

Now we have all the elements needed for proof of the Theorem 3.1.

Proof of Theorem 3.1:[Rank-1 case]

Proof. Lemma 3.1 has shown $\hat{u}^0$ satisfies the row norm bounds condition. From Lemma 3.2 we get $dist(v^{t+1}, v^*) \le \frac{1}{2}dist(u^t, u^*) + 5\delta\|M - M_1\|_F/\sigma^*$. Hence $dist(v^{t+1}, v^*) \le \frac{1}{4^t}dist(\hat{u}^0, u^*) + 10\delta\|M - M_1\|_F/\sigma^*$. After $t = O(\log(\frac{1}{\zeta}))$ iterations we get $dist(v^{t+1}, v^*) \le \zeta + 10\delta\|M - M_1\|_F/\sigma^*$ and $dist(u^t, u^*) \le \zeta + 10\delta\|M - M_1\|_F/\sigma^*$.

Hence,

$$\|M_1 - \hat{u}^t(\hat{v}^{t+1})^T\|$$
$$\le \|(I - u^t(u^t)^T)M_1\| + \|u^t\left((u^t)^T M_1 - (v^{t+1})^T\right)\|$$
$$\overset{\zeta_1}{\le} \sigma_1^* dist(u^t, u^*) + \|\sigma^* B^{-1}(\langle u^t, u^*\rangle B - C)v^*\|$$
$$\quad + \|B^{-1}y\|$$
$$\overset{\zeta_2}{\le} \sigma_1^* dist(u^t, u^*) + 2\delta_3\|M\|dist(u^t, u^*)$$
$$\quad + 2dist(u^t, u^*)\|M - M_1\| + 2\delta\|M - M_1\|_F$$
$$\le c\sigma_1^*\zeta + \epsilon\|M - M_1\|_F.$$

ζ_1 follows from equation (B.11) and ζ_2 from $\|B^{-1}\| \le \frac{1}{1-\delta_3} \le 2$ from Lemma B.3.

From Lemma B.2 we have, in each iteration with probability greater than $1 - \frac{\gamma}{T\log(n)}$ we have $\|(u^t)^T R_\Omega(M - M_1)\| \le dist(u^t, u^*)\|M - M_1\| + \delta\|M - M_1\|_F$. Hence the probability of failure in T iterations is less than γ.

B.3 Rank-r proofs: Let SVD of M_r be $U^*\Sigma^*(V^*)^T$, U^*, V^* are $n \times r$ orthonormal matrices and Σ^* is a $r \times r$ diagonal matrix with $\Sigma_{ii}^* = \sigma_i^*$.

In this section we will present rank-r proof of Lemma 3.2. Before that we will present rank-r version of the supporting lemmas.

Now like shown in [19], we will analyze a equivalent algorithm to algorithm 2 where the iterates are orthogonalized at each step. This makes analysis significantly simpler to present. Let, $\hat{U}^{(t)} = U^{(t)}R^{(t)}$ and $\hat{V}^{(t+1)} = V^{(t+1)}R^{(t+1)}$ be the respective QR factorizations. Then we replace step 7 of the algorithm 2 with

$$\hat{V}^{(t+1)} = \arg\min_{V \in \mathbb{R}^{n \times r}} \|R_{\Omega_{2t+1}}(M - U^{(t)}V^T)\|_F^2.$$

We similarly change step 8 too.

We also assume that samples for each iteration are generated independently. For simplicity we will drop the subscripts on Ω that denote different set of samples in each iteration in the rest of the proof. The weighted alternating minimization updates at the $t+1$ iteration are,
(B.20)
$$(\hat{V}^{(t+1)})^j = (B^j)^{-1}\left(C^j\Sigma^*(V^*)^j + (U^{(t)})^T R_\Omega(M - M_r)_j\right),$$

where B^j and C^j are $r \times r$ matrices.

$$B^j = \sum_i \delta_{ij}w_{ij}(U^{(t)})^i(U^{(t)})^{iT},$$
$$C^j = \sum_i \delta_{ij}w_{ij}(U^{(t)})^i(U^*)^{iT}.$$

Writing in terms of power method updates we get,

(B.21)
$$(\widehat{V}^{(t+1)})^j = \left((U^{(t)})^T U^* - (B^j)^{-1}(B^j (U^{(t)})^T U^* \right.$$
$$\left. -C^j)\right)\Sigma^*(V^*)^j + (B^j)^{-1}(U^{(t)})^T R_\Omega (M - M_r)_j.$$

Hence,

$$(\widehat{V}^{(t+1)})^T = (U^{(t)})^T U^* \Sigma^*(V^*)^T - F$$
$$+ \sum_{j=1}^{n} (B^j)^{-1}(U^{(t)})^T R_\Omega (M - M_r)_j e_j^T,$$

where the jth column of F, $F_j = \left((B^j)^{-1}(B^j(U^{(t)})^T U^* - C^j)\right)\Sigma^*(V^*)^j$.

First we will bound $\|B^j\|$ using matrix Bernstein inequality.

LEMMA B.5. *For Ω generated according to (3.2) and under the assumptions of Lemma 3.2 the following holds:*

(B.22) $\|B^j - I\| \le \delta_2$, *and* $\|C^j - (U^{(t)})^T U^*\| \le \delta_2$,

with probability greater that $1 - \frac{2}{n^2}$, for $m \ge \beta n r \kappa \log(n)$, $\beta \ge \frac{4*48c_1^2}{\delta_2^2}$ and $\delta_2 \le 3r$.

Now we will bound the error caused by the $M - M_r$ component in each iteration.

LEMMA B.6. *For Ω generated according to (3.2) and under the assumptions of Lemma 3.2 the following holds:*

(B.23) $\left\|(U^{(t)})^T R_\Omega (M - M_r) - (U^{(t)})^T (M - M_r)\right\|$
$$\le \delta \|M - M_r\|_F,$$

with probability greater that $1 - \frac{1}{c_2 \log(n)}$, for $m \ge \beta n r \log(n)$, $\beta \ge \frac{4c_1^2 c_2}{\delta^2}$. Hence,$\left\|(U^{(t)})^T R_\Omega (M - M_r)\right\| \le dist(U^{(t)}, U^*)\|M - M_r\| + \delta \|M - M_r\|_F$, for constant δ.

Since $\|B^j\|$ is bounded by the previous lemma, to get a bound on the norm of the error term in equation (B.21), $\|F\|$, we need to bound $\|\tilde{F}\|$, where $\tilde{F}_j = (B^j(U^{(t)})^T U^* - C^j)\Sigma^*(V^*)^j$.

LEMMA B.7. *For Ω generated according to (3.2) and under the assumptions of Lemma 3.2 the following holds:*

(B.24) $\|\tilde{F}\| \le \delta_2 \sigma_1^* dist(U^{(t)}, U^*)$

with probability greater that $1 - \frac{2}{n^2}$, for $m \ge \beta n \log(n)$, $\beta \ge \frac{32 r^3 \kappa^2}{\delta_2^2}$, $c_1 \le 8\kappa\sqrt{r}$ and $\delta_2 \le \frac{1}{2}$.

The proofs of the above three lemmas are similar to the rank-1 proofs presented in the previous section and we skip them in this version.

Now since $\widehat{V}^{(t+1)} = V^{(t+1)} R^{(t+1)}$,

(B.25) $\sigma_{\min}(R^{(t+1)}) = \sigma_{\min}(\widehat{V}^{(t+1)})$
$$\overset{\zeta_1}{\ge} \sigma_{\min}((U^{(t)})^T U^* \Sigma^*(V^*)^T) - \|F\|$$
$$- \|\sum_{j=1}^{n}(B^j)^{-1}(U^{(t)})^T R_\Omega (M - M_r)_j e_j^T\|.$$

ζ_1 follows from (B.21).

Now $\sigma_{\min}((U^{(t)})^T U^* \Sigma^*(V^*)^T) \ge \sigma_r^* \sqrt{1 - dist(U^{(t)}, U^*)^2}$. Hence, $\|F\| \le \frac{1}{1-\delta_2}\|\tilde{F}\| \le \frac{\delta_2}{1-\delta_2}\sigma_1^* dist(U^{(t)}, U^*)$.

$$\|\sum_{j=1}^{n}(B^j)^{-1}(U^{(t)})^T R_\Omega (M - M_r)_j e_j^T\|$$
$$\le \frac{1}{1-\delta_2}\|(U^{(t)})^T R_\Omega (M - M_r)\|_F$$
$$\le \frac{1}{1-\delta_2}dist(U^{(t)}, U^*)\|M - M_r\|$$
$$+ \frac{\delta}{1-\delta_2}\|M - M_r\|_F,$$

from Lemma B.6.

Hence,

$$\sigma_{\min}(R^{(t+1)}) \ge \sigma_r^* \left(\sqrt{1 - dist(U^{(t)}, U^*)^2} \right.$$
$$\left. -\kappa\frac{\delta_2}{1-\delta_2}dist(U^{(t)}, U^*) - \frac{2}{1-\delta_2}\|M - M_r\|_F / \sigma_r^* \right)$$
$$\ge \frac{\sigma_r^*}{2},$$

for enough number of samples m.

Now we are ready to present proof of Lemma 3.2 for rank-r case.

Proof of Lemma 3.2:

Proof. The proof like in rank-1 case has two steps. In the first step we show that $dist(V^{(t+1)}, V^*)$ decreases in each iteration. In the second step we show row norm bounds. Recall from the assumptions of the Lemma we have the following row norm bounds for $U^{(t)}$;

(B.26)
$$\|(U^{(t)})^i\| \le c_1 \sqrt{\|M^i\|^2 / \|M\|_F^2 + |M_{ij}|/\|M\|_F}, \quad \forall i.$$

We will prove that $V^{(t+1)}$ satisfies this condition.

Bounding $dist(V^{(t+1)}, V^*)$:

$$dist(V^{(t+1)}, V^*) = \|(V^{(t+1)})^T V^*_\perp\|$$

$$\overset{\zeta_1}{\leq} \|(R^{(t+1)})^{-1^T} F V^*_\perp\|$$

$$+ \frac{1}{1-\delta_2} \|(R^{(t+1)})^{-1^T}(U^{(t)})^T R_\Omega(M - M_r)V^*_\perp\|_F$$

$$\overset{\zeta_2}{\leq} \frac{1}{\sigma_{\min}(R^{(t+1)})} \left(\|F\| + \frac{1}{1-\delta_2} dist(U^{(t)}, U^*)\|M - M_r\| \right.$$

$$\left. + \frac{\delta}{1-\delta_2} \|M - M_r\|_F \right)$$

$$\leq \frac{2}{\sigma_r^*} \left(\frac{\delta_2}{1-\delta_2} \|M\| dist(U^{(t)}, U^*) \right.$$

$$\left. + \frac{1}{1-\delta_2} dist(U^{(t)}, U^*)\|M - M_r\| + \frac{\delta}{1-\delta_2} \|M - M_r\|_F \right)$$

$$\leq \frac{1}{2} dist(U^{(t)}, U^*) + 5\delta \|M - M_r\|_F / \sigma_r^*,$$

for $\delta_2 \leq \frac{1}{16\kappa}$. ζ_1 follows from (B.21). ζ_2 follows from Lemma B.6.

Bounding $\|(V^{(t+1)})^j\|$:

From Lemma B.5 and (B.26) we get that $\sigma_{\min}(B^j) \geq 1 - \delta_2$ and $\sigma_{\max}(C^j) \leq 1 + \delta_2$. Recall that (B.27)

$$(V^{(t+1)})^j = (R^{(t+1)})^{-1^T} \left((B^j)^{-1}(U^{(t)})^T R_\Omega(M)_j \right).$$

Hence,

$$\|(V^{(t+1)})^j\| \leq \frac{1}{\sigma_{\min}(R^{(t+1)})} \left(\frac{1}{1-\delta_2} \|(U^{(t)})^T R_\Omega(M)_j\| \right).$$

We will bound $\|(U^{(t)})^T R_\Omega(M)_j\|$ using matrix Bernstein inequality.

Let $X_i = (\delta_{ij} - \hat{q}_{ij})w_{ij}M_{ij}(U^{(t)})^i e_j^T$. Then $\mathbb{E}[X_i] = 0$ and $\sum_i X_i = (U^{(t)})^T R_\Omega(M)_j - (U^{(t)})^T M_j$. Now $\|X_i\| \leq \frac{c_1 2n}{m}\sqrt{|M_{ij}|\|M\|_F}$ and $\|\mathbb{E}[\sum_i X_i X_i^T]\| \leq \frac{8c_1^2 n}{m}\|M_j\|^2$. Hence applying matrix Bernstein inequality with $t = \delta_2\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}$, implies

$$\|(U^{(t)})^T R_\Omega(M)_j\| \leq \|M_j\| + \delta_2\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F}$$

$$\leq (1+\delta_2)\sqrt{\|M_j\|^2 + |M_{ij}|\|M\|_F},$$

with probability greater than $1 - \frac{2}{n^2}$ for $m \geq \frac{24c_1^2}{\delta_2^2}n\log(n)$. Hence $\|(V^{(t+1)})^j\| \leq 8\kappa\sqrt{r}\sqrt{\frac{\|M_j\|^2}{\|M\|_F^2} + \frac{|M_{ij}|}{\|M\|_F}}$.

Hence we have shown that $(V^{(t+1)})^j$ satisfies corresponding row norm bounds. This completes the proof.

Now we have all the elements needed for proof of the Theorem 3.1. The proof is similar to the proof in the rank-1 case and follows from the Lemmas 3.1 and 3.2

C Proofs of section 3.3

We will now discuss proof of Theorem 3.2. The proof follows same structure as proof of Theorem 3.1 with few key changes because of the absence of $L1$ term in the sampling and the special structure of $M = AB$. Again for simplicity we will present proofs only for the case of $n_1 = n_2 = n$.

Recall that $\hat{q}_{ij} = \min(1, q_{ij})$ where $q_{ij} = m \cdot \left(\frac{\|A^i\|^2}{n\|A\|_F^2} + \frac{\|B_j\|^2}{n\|B\|_F^2} \right)$. Also, let $w_{ij} = 1/\hat{q}_{ij}$.

First we will abstract out the properties of the sampling distribution (3.5) that we use in the rest of the proof. Also let $C_{AB} = \frac{(\|A\|_F^2 + \|B\|_F^2)^2}{\|AB\|_F^2}$

LEMMA C.1. *For Ω generated according to (3.5) and under the assumptions of Lemma C.2 the following holds, for all (i, j) such that $q_{ij} \leq 1$.*

$$(C.28) \qquad \frac{M_{ij}}{\hat{q}_{ij}} \leq \frac{n}{2m}(\|A\|_F^2 + \|B\|_F^2),$$

$$(C.29) \qquad \sum_{\{j:\hat{q}_{ij}=q_{ij}\}} \frac{M_{ij}^2}{\hat{q}_{ij}} \leq \frac{n}{m}(\|A\|_F^2 + \|B\|_F^2)^2,$$

$$(C.30) \qquad \frac{\|(U^*)^i\|^2}{\hat{q}_{ij}} \leq \frac{n}{m}\frac{(\|A\|_F^2 + \|B\|_F^2)^2}{\|A \cdot B\|_F^2},$$

and

$$(C.31) \qquad \frac{\|(U^*)^i\|\|(V^*)^j\|}{\hat{q}_{ij}} \leq \frac{n}{m}\frac{(\|A\|_F^2 + \|B\|_F^2)^2}{\|A \cdot B\|_F^2}.$$

The proof of the lemma C.1 is straightforward from the definition of q_{ij}.

Now, similar to proof of Theorem 3.1, we divide our analysis in two parts: initialization analysis and weighted alternating minimization analysis.

C.1 Initialization

LEMMA C.2. (INITIALIZATION) *Let the set of entries Ω be generated according to $\hat{q}_{ij}$ (3.5). Also, let $m \geq CC_{AB}\frac{n}{\delta^2}\log(n)$. Then, the following holds (w.p. $\geq 1 - \frac{2}{n^{10}}$):*

$$(C.32) \qquad \|R_\Omega(AB) - AB\| \leq \delta\|AB\|_F.$$

Also, if $\|AB - (AB)_r\|_F \leq \frac{1}{576\kappa r^{1.5}}\|(AB)_r\|_F$, then the following holds (w.p. $\geq 1 - \frac{2}{n^{10}}$):

$$\|(\widehat{U}^{(0)})^i\| \leq 8\sqrt{r}\sqrt{\|A^i\|^2/\|A\|_F^2} \quad and \quad dist(\widehat{U}^{(0)}, U^*) \leq \frac{1}{2},$$

where $\widehat{U}^{(0)}$ is the initial iterate obtained using Steps 4, 5 of Sub-Procedure 2. $\kappa = \sigma_1^*/\sigma_r^*$, σ_i^* is the i-th singular value of AB, $(AB)_r = U^*\Sigma^*(V^*)^T$.

Proof. First we show that $R_\Omega(AB)$ is a good approximation of AB.

Let $M = AB$. We prove this part of the lemma using the matrix Bernstein inequality. Let $X_{ij} = (\delta_{ij} - \hat{q}_{ij})w_{ij}M_{ij}e_i e_j^T$. Note that $\{X_{ij}\}_{i,j=1}^n$ are independent zero mean random matrices. Also $R_\Omega(AB) - \mathbb{E}[R_\Omega(AB)] = \sum_{ij} X_{ij}$.

First we will bound $\|X_{ij}\|$. When $mq_{ij} \geq 1$, $\hat{q}_{ij} = 1$ and $\delta_{ij} = 1$, and $X_{ij} = 0$ with probability 1. Hence we only need to consider cases when $\hat{q}_{ij} = mq_{ij} \leq 1$. We will assume this in all the proofs without explicitly mentioning it any more.

$$\|X_{ij}\| = \max\{|(1-\hat{q}_{ij})w_{ij}M_{ij}|, |\hat{q}_{ij}w_{ij}M_{ij}|\}.$$

Recall $w_{ij} = 1/\hat{q}_{ij}$. Hence

$$|(1-\hat{q}_{ij})w_{ij}M_{ij}| \leq \left|\frac{M_{ij}}{\hat{q}_{ij}}\right| \overset{\zeta_1}{\leq} \frac{n}{2m}(\|A\|_F^2 + \|B\|_F^2).$$

ζ_1 follows from (C.28).

$$|\hat{q}_{ij}w_{ij}M_{ij}| = |M_{ij}| \overset{\zeta_1}{\leq} \left|\frac{M_{ij}}{\hat{q}_{ij}}\right| \leq \frac{n}{2m}(\|A\|_F^2 + \|B\|_F^2).$$

ζ_1 follows from $\hat{q}_{ij} \leq 1$.

Hence, $\|X_{ij}\|$ is bounded by $L = \frac{n}{2m}(\|A\|_F^2 + \|B\|_F^2)$. Recall that this is the step in the proof of Lemma 3.1 that required the L1 term in sampling, which we didn't need now because of the structure AB of the matrix. Now we will bound the variance. Recall,

$$\left\|\mathbb{E}\left[\sum_{ij}X_{ij}X_{ij}^T\right]\right\| = \max_i \left|\sum_j \hat{q}_{ij}(1-\hat{q}_{ij})w_{ij}^2 M_{ij}^2\right|.$$

Now,

$$\sum_j \hat{q}_{ij}(1-\hat{q}_{ij})w_{ij}^2 M_{ij}^2 \leq \sum_j \frac{M_{ij}^2}{(\hat{q}_{ij})} \overset{\zeta_1}{\leq} \frac{n}{m}(\|A\|_F^2 + \|B\|_F^2)^2.$$

ζ_1 follows from (C.29). Hence

$$\left\|\mathbb{E}\left[\sum_{ij}X_{ij}X_{ij}^T\right]\right\| \leq \frac{n}{m}(\|A\|_F^2 + \|B\|_F^2)^2.$$

We can prove the same bound for the $\left\|\mathbb{E}\left[\sum_{ij}X_{ij}^T X_{ij}\right]\right\|$. Hence $\sigma^2 = \frac{n}{m}(\|A\|_F^2 + \|B\|_F^2)^2$. Now using matrix Bernstein inequality with $t = \delta\|AB\|_F$ gives, with probability $\geq 1 - \frac{2}{n^2}$,

$$\|R_\Omega(AB) - \mathbb{E}[R_\Omega(AB)]\| = \|R_\Omega(AB) - AB\|$$
$$\leq \delta\|AB\|_F.$$

Once we have $\|R_\Omega(M) - M\| \leq \delta\|M\|_F$ proof that trimming step guarantees

$$\|(\widehat{U}^{(0)})^i\| \leq 8\sqrt{r}\sqrt{\frac{\|A^i\|^2}{\|A\|_F^2}} \quad \text{and} \quad dist(\widehat{U}^{(0)}, U^*) \leq \frac{1}{2},$$

follows from the same argument as in Lemma 3.1.

C.2 Weighted AltMin Analysis

LEMMA C.3. (WALTMIN DESCENT) *Let hypotheses of Theorem 3.2 hold. Also, let $\|AB - (AB)_r\|_F \leq \frac{1}{576\kappa r\sqrt{r}}\|(AB)_r\|_F$. Let $\widehat{U}^{(t)}$ be the t-th step iterate of Sub-Procedure 2 (called from $WAltMin(P_\Omega(A \cdot B), \Omega, \hat{q}, T)$), and let $\widehat{V}^{(t+1)}$ be the $(t+1)$-th iterate (for V). Also, let $\|(U^{(t)})^i\| \leq 8\sqrt{r}\kappa\sqrt{\|A^i\|^2/\|A\|_F^2}$ and $dist(\widehat{U}^{(t)}, U^*) \leq \frac{1}{2}$, where $U^{(t)}$ is a set of orthonormal vectors spanning $\widehat{U}^{(t)}$. Then, the following holds (w.p. $\geq 1 - \gamma/T$):*

$$dist(\widehat{V}^{(t+1)}, V^*) \leq \frac{1}{2}dist(\widehat{U}^{(t)}, U^*) + \epsilon\frac{\|AB - (AB)_r\|_F}{\sigma_r^*},$$

and $\|(V^{(t+1)})^j\| \leq 8\sqrt{r}\kappa\sqrt{\|B_j\|^2/\|B\|_F^2}$, where $V^{(t+1)}$ is a set of orthonormal vectors spanning $\widehat{V}^{(t+1)}$.

For the sake of simplicity we will discuss the proof for rank-1($r = 1$) case for this part of the algorithm. Rank-r proof follows by combining the below analysis with rank-r analysis of Lemma 3.2 (see Section B.3). Before presenting the proof of this Lemma, we will state couple of supporting lemmas. The proofs of these supporting lemmas follows very close to the ones in section B.2.

LEMMA C.4. *For Ω sampled according to (3.5) the following holds:*

$$(C.33) \qquad \left|\sum_j \delta_{ij}w_{ij}(u_j^*)^2 - \sum_j (u_j^*)^2\right| \leq \delta_1,$$

with probability greater that $1 - \frac{2}{n^2}$, for $m \geq \beta C_{AB}n\log(n)$, $\beta \geq \frac{16}{\delta_1^2}$ and $\delta_1 \leq 3$.

We assume that samples for each iteration are generated independently. For simplicity we will drop the subscripts on Ω that denote different set of samples in each iteration in the rest of the proof. The weighted alternating minimization updates at the $t + 1$ iteration are,

(C.34)
$$\|\hat{u}^t\|\widehat{v}_j^{t+1} = \sigma^* v_j^* \frac{\sum_i \delta_{ij}w_{ij}u_i^t u_i^*}{\sum_i \delta_{ij}w_{ij}(u_i^t)^2} + \frac{\sum_i \delta_{ij}w_{ij}u_i^t(M - M_1)_{ij}}{\sum_i \delta_{ij}w_{ij}(u_i^t)^2}.$$

Writing in terms of power method updates we get,
(C.35)
$$\|\widehat{u}^t\|\widehat{v}^{t+1} = \sigma^*\langle u^*, u^t\rangle v^* - \sigma^* P^{-1}(\langle u^t, u^*\rangle P - Q)v^* + P^{-1}y,$$

where P and Q are diagonal matrices with $P_{jj} = \sum_i \delta_{ij}w_{ij}(u_i^t)^2$ and $Q_{jj} = \sum_i \delta_{ij}w_{ij}u_i^t u_i^*$ and y is the vector $R_\Omega(M-M_1)^T u^t$ with entries $y_j = \sum_i \delta_{ij}w_{ij}u_i^t(M-M_1)_{ij}$.

Now we will bound the error caused by the $M - M_r$ component in each iteration.

LEMMA C.5. *For Ω generated according to (3.5) the following holds:*
(C.36)
$$\left\|(U^{(t)})^T R_\Omega(M-M_r) - (U^{(t)})^T(M-M_r)\right\| \le \delta\|M-M_r\|_F$$

with probability greater that $1 - \frac{1}{c_2\log(n)}$, *for* $m \ge \beta nr\log(n)$, $\beta \ge \frac{4c_1^2 c_2}{\delta^2}$. *Hence,* $\left\|(U^{(t)})^T R_\Omega(M-M_r)\right\| \le dist(U^{(t)}, U^*)\|M - M_r\| + \delta\|M-M_r\|_F$, *for constant δ.*

LEMMA C.6. *For Ω sampled according to (3.5) and for u^t satisfying equation (C.38) the following holds:*

(C.37)
$$\|(\langle u^t, u^*\rangle P - Q)v^*\| \le \delta_1\sqrt{1 - \langle u^*, u^t\rangle^2},$$

with probability greater than $1 - \frac{2}{n^2}$, *for* $m \ge \beta C_{AB}n\log(n), \beta \ge \frac{48c_1^2}{\delta_1^2}$ *and $\delta_1 \le 3$.*

Now we will provide proof of lemma C.3.
Proof of lemma C.3:[Rank-1 case]

Proof. Let u^t and v^{t+1} be the normalized vectors of the iterates $\widehat{u}^t$ and $\widehat{v}^{t+1}$. In the first step we will prove that the distance between u^t, v^{t+1} and u^*, v^* decreases with each iteration. From the assumptions of the lemma we have;

(C.38)
$$|u_i^t| \le c_1\sqrt{\|A^i\|^2/\|A\|_F^2}.$$

In the second step we will prove that v^{t+1} satisfies $|v_j^{t+1}| \le c_1\sqrt{\|B_j\|^2/\|B\|_F^2}$.

Bounding $\langle v^{t+1}, v^*\rangle$: The proof of this part is same as in the proofs in previous section and hence we skip it.

Bounding v_j^{t+1}:
From Lemma C.4 and (C.38) we get that $\left|\sum_i \delta_{ij}w_{ij}(u_i^t)^2 - 1\right| \le \delta_1$ and $\left|\sum_i \delta_{ij}w_{ij}u_i^* u_i^t - \langle u^*, u^t\rangle\right| \le \delta_1$, when $\beta \ge \frac{16c_1^2}{\delta_1^2}$. Hence,

(C.39)
$$1 - \delta_1 \le P_{jj} = \sum_i \delta_{ij}w_{ij}(u_i^t)^2 \le 1 + \delta_1,$$

and
(C.40)
$$Q_{jj} = \sum_i \delta_{ij}w_{ij}u_i^t u_i^* \le \langle u^t, u^*\rangle + \delta_1.$$

Recall that
$$\|\widehat{u}^t\|\left|\widehat{v}_j^{t+1}\right| = \left|\frac{\sum_i \delta_{ij}w_{ij}u_i^t M_{ij}}{\sum_i \delta_{ij}w_{ij}(u_i^t)^2}\right| \le \frac{1}{1-\delta_1}\sum_i \delta_{ij}w_{ij}u_i^t M_{ij}.$$

We will bound using $\sum_i \delta_{ij}w_{ij}u_i^t M_{ij}$ using bernstein inequality. Let $X_i = (\delta_{ij} - \widehat{q}_{ij})w_{ij}u_i^t M_{ij}$. Then $\sum_i \mathbb{E}[X_i] = 0$ and $\sum_i u_i^t M_{ij} \le \|M_j\|$ by Cauchy-Schwartz inequality. $\sum_i \text{Var}(X_i) = \sum_i \widehat{q}_{ij}(1 - \widehat{q}_{ij})(w_{ij})^2(u_i^t)^2 M_{ij}^2 \le \sum_i w_{ij}(u_i^t)^2 M_{ij}^2 \le \frac{nc_1^2}{m}\|M_j\|^2 \le \frac{nc_1^2}{m}\|B_j\|^2\|A\|_F^2$. Finally $|X_{ij}| \le |w_{ij}u_i^t M_{ij}| \le \frac{nc_1}{m}\|A\|_F\|B_j\|$. Hence applying bernstein inequality with $t = \delta\frac{\|B_j\|}{\|B\|_F}\|AB\|_F$ gives, $\sum_i \delta_{ij}w_{ij}u_i^t M_{ij} \le (1 + \delta_1)\frac{\|B_j\|}{\|B\|_F}\|AB\|_F$ with probability greater than $1 - \frac{2}{n^3}$ when $m \ge \frac{24c_1^2}{\delta_1^2}C_{AB}n\log(n)$. For $\delta_1 \le \frac{1}{20}$, we get, $\|\widehat{u}^t\|\left|\widehat{v}_j^{t+1}\right| \le \frac{21}{19}\frac{\|B_j\|}{\|B\|_F}\|AB\|_F$.

Now we will bound $\|\widehat{v}^{t+1}\|$.

$$\|\widehat{u}^t\|\|\widehat{v}^{t+1}\| \ge \|\widehat{u}^t\|\langle \widehat{v}^{t+1}, v^*\rangle$$
$$\overset{\zeta_1}{\ge} \sigma^*\langle u^t, u^*\rangle - \sigma^*\frac{\delta_1}{1-\delta_1}\sqrt{1 - \langle u^*, u^t\rangle^2} - \frac{1}{1-\delta_1}\|y^T v^*\|$$
$$\overset{\zeta_2}{\ge} \sigma^*\langle u^0, u^*\rangle - 2\sigma^*\delta_1\sqrt{1 - \langle u^*, u^0\rangle^2} - 2\delta\|M - M_1\| \overset{\zeta_3}{\ge} \frac{2}{5}\sigma^*.$$

ζ_1 follows from Lemma C.6 and equations (C.35) and (C.39). ζ_2 follows from using $\langle u^*, u^0\rangle \le \langle u^*, u^t\rangle$ and $\delta_1 \le \frac{1}{20}$. ζ_3 follows by initialization and using the assumption on m with large enough $C > 0$. Hence we get

$$v_j^{t+1} = \frac{\widehat{v}_j^{t+1}}{\|\widehat{v}^{t+1}\|} \le 3\frac{21}{19}\frac{\|B_j\|}{\|B\|_F}\frac{\|AB\|_F}{\sigma^*} \le c_1\frac{\|B_j\|}{\|B\|_F},$$

for $c_1 = 6$.

Hence we have shown that v^{t+1} satisfies the row norm bounds. This completes the proof.

The proof of the Theorem 3.2 now follows from the Lemma C.2 and Lemma C.3 similar to the proof of Theorem 3.1.

Algorithm 3 Distributed low rank approximation algorithm

---

input Matrix $M_{\mathbf{r_k}}$ at server k, rank-r, number of samples m and number of iterations T.

1: *Sampling:* Each server k computes column norms of $M_{\mathbf{r_k}}$, $\|M_{\mathbf{r_k}}\|_{1,1}$ and communicates to CP. CP computes column norms of M, $\|M\|_{1,1}$, $\|M\|_F$ and communicates to all servers.

2: Each server k samples (i,j)th entry with probability $\min\{m(\frac{\|M_{\mathbf{r_k}}^i\|^2 + \|M_j\|^2}{2n\|M\|_F^2} + \frac{(M_{\mathbf{r_k}})_{ij}}{\|M\|_{1,1}}), 1\}$ for rows $\{r_k\}$ and generates Ω^k.

3: Each server k sends lists of columns $(\{c_k\} \subset [d])$ where $R_{\Omega^k}(M_{\mathbf{r_k}})$ has sampled entries, to CP.

4: *Initialization:* CP generates random $n \times r$ matrix $Y^{(0)}$ and communicates $Y_{\mathbf{c_k}}^{(0)}$ to server k.

5: **for** $t = 0$ to $\log(1/c)$ **do**

6:　　Each server k computes $\widehat{Y}_{\mathbf{c_k}}^{(t+1)} = R_{\Omega^k}(M_{\mathbf{r_k}})^T R_{\Omega^k}(M_{\mathbf{r_k}}) * Y_{\mathbf{c_k}}^{(t)}$ and communicates to CP.

7:　　CP computes $Y^{(t+1)} = \sum_k \widehat{Y}_{\mathbf{c_k}}^{(t+1)}$, normalize and communicates $Y_{\mathbf{c_k}}^{(t+1)}$ to server k.

8: **end for**

9: *WAltMin:* Each server k set $\widehat{V}_{\mathbf{c_k}}^{(0)} = Y_{\mathbf{c_k}}^{(t+1)}$.

10: **for** $t = 0$ to $T - 1$ **do**

11:　　Each server k computes $(\widehat{U}^{(t+1)})^i = \arg\min_{x \in \mathbb{R}^r} \sum_{j:(i,j)\in\Omega^k} w_{ij} \left(M_{ij} - x^T (\widehat{V}^t)^j \right)^2$, for all $i \in \{r_k\}$.

12:　　Each server k sends to CP; $z_{kj} = \left(\widehat{U}_{\mathbf{r_k}}^{(t+1)}\right)^T R_{\Omega^k}(M_{\mathbf{r_k}})e_j$ and $B_{kj} = \sum_{i:(i,j)\in\Omega^k} w_{ij} u u^T$, $u = (\widehat{U}^{(t+1)})^i$ for all $j \in \{c_k\}$.

13:　　CP computes $B_j = \sum_k B_{kj}$ and $(\widehat{V}^{(t+1)})^j = B_j^{-1} \sum_k z_{kj}$ for $j = 1, \cdots, d$ and communicates $(\widehat{V}^{(t+1)})_{\mathbf{c_k}}$ to server k.

14: **end for**

output Server k has $\widehat{U}_{\mathbf{r_k}}^{(t+1)}$ and CP has $\widehat{V}^{(t+1)}$.