Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions

1 Summary of the paper

1.1 Main Procedure to decompose a objective matrix

1.1.1 Main steps for the procedure

: The task of computing a low rank approximation to a given matrix can be split into two comutational stages.
Stage A: constructing a low-dimensional subspace that capture the action of the matrix(finding a matrix $\mathbf{Q}$)

$$\mathbf{Q} \text{ has orthonormal columns and } \mathbf{A} \approx \mathbf{Q}\mathbf{Q}^*\mathbf{A}$$

Stage B: restricting the matrix to the subspace and then compute standard factorization of the reduced matrix.

1.1.2 Step A, Proto-Algorithm: solving the fixed rank problem and finding Q

:

1. Given an $m \times n$ matrix $\mathbf{A}$, a target rank k, and an oversampling parameter p, this procedure computes an $m \times (k+p)$ matrix $\mathbf{Q}$ whose columns are orthonormal and whose range approximates the range of $\mathbf{A}$
   (a) Draw a random $n \times (k+p)$ test matrix Ω
   (b) Form the matrix product $\mathbf{Y} = \mathbf{A\Omega}$
   (c) Construct a matrix $\mathbf{Q}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$
2. We can make this more concrete: Given an $m \times n$ matirx $\mathbf{A}$, a tolerance ϵ, and an integer r, the following scheme computes an orthonormal matrix $\mathbf{Q}$ such that $\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}\| \leq \epsilon$ holds with probability at least $1 - \min_{m,n} 10^{-r}$

1. Draw sandard Gaussian vector $w^{(1)}, \cdots, w^{(r)}$ of length n
2. For $i = 1, 2, \cdots, r$ compute $\mathbf{y}^{(i)} = \mathbf{A}\mathbf{w}^{(i)}$
3. j = 0
   $\mathbf{Q}^{(0)} = [\ ]$, the $m \times 0$ empty matrix.
4. **While** $\max\{\|\mathbf{y}^{(j+1)}\|, \|\mathbf{y}^{(j+2)}\|, \cdots, \|\mathbf{y}^{(j+r)}\|\} > \epsilon/(10/\sqrt{2/\pi})$
   j = j+1
   Overwrite $\mathbf{y}^{(j)}$ by $(\mathbf{I} - \mathbf{Q}^{(\mathbf{j-1})}(\mathbf{Q}^{(\mathbf{j-1})})^*)\mathbf{y}^{(j)}$
   $\mathbf{q}^{(j)} = \mathbf{y}^{(j)}/\|\mathbf{y}^{(j)}\|$
   $\mathbf{Q}^{(j)} = [\mathbf{Q}^{(j-1)}\mathbf{q}^{(j)}]$
   Draw a standard Gaussian vector $\mathbf{w}^{(j+r)}$ of length n
   $\mathbf{y}^{(j+r)} = (\mathbf{I} - \mathbf{Q}^{(\mathbf{j})}(\mathbf{Q}^{(\mathbf{j})})^*)\mathbf{A}\mathbf{w}^{(j+r)}$
   for $i = (j+1), \cdots, (j+r-1)$,
   Overwrite $\mathbf{y}^{(i)}$ by $\mathbf{y}^{(i)} - \mathbf{q}^{(j)}\langle \mathbf{q}^{(j)}, \mathbf{y}^{(i)} \rangle$
   end for
   end while
5. $\mathbf{Q} = \mathbf{Q}^{(j)}$

1.1.3 Step B, Construction of standard factorizations.

1. Factorizations based on forming $\mathbf{Q}^*\mathbf{A}$ directly : You can check 1.4. Stage B part.
2. Postprocessing via row extraction: Give a matrix $\mathbf{Q}$, we can obtain a rank-k factorization

$$\mathbf{A} \approx \mathbf{XB}$$

Where $\mathbf{Q} = \mathbf{XQ}_{(\mathbf{J},:)}$. Combining above results, we can get $\mathbf{A} \approx \mathbf{QQ}^*\mathbf{A} = \mathbf{XQ}_{(\mathbf{J},:)}\mathbf{Q}^*\mathbf{A} \rightarrow \mathbf{A}_{(\mathbf{J},:)} \approx \mathbf{Q}_{(\mathbf{J},:)}\mathbf{Q}^*\mathbf{A} \rightarrow \mathbf{B} = \mathbf{A}_{(\mathbf{J},:)}$

1.1.4 Example: Prototype for randomized SVD (singular values slowly decaying case)

Given an $m \times n$ matrix $\mathbf{A}$, a target number k of singular vectors, and an exponent q, this procedure computes an approximate rank-2k factorization$\mathbf{U\Sigma V}^*$

Stage A

1. Generate an $n \times 2k$ Gaussian test matrix Ω
2. Form $\mathbf{Y} = (\mathbf{AA}^*)^{\mathbf{q}}\mathbf{A\Omega}$
3. Construct a matrix $\mathbf{Q}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$

Stage B

1. Form $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$
2. Compute an SVD of the small matrix: $\mathbf{B} = \tilde{\mathbf{U}}\mathbf{\Sigma V}^*$
3. Set $\mathbf{U} = \mathbf{Q}\tilde{\mathbf{U}}$

1.2 Theoretical assurance

1.2.1 Error bounds before drawing test random matrices

Theorem 1. *$A = U\Sigma V^*$ and $Y = A\Omega$, and assume $\Omega_1$has full rank, then*

$$\|(I - P_Y)A\|^2 \leq \|\Sigma_2\|^2 + \|\Sigma_2\Omega_2\Omega_1^-\|^2$$

We can expend this and can get the power scheme's bound.

Theorem 2. *Let $A = (BB^*)^q A$ and compute the sample matirx $Z = B\Omega$*

$$\|(I - P_z)A\| \leq \|(I - P_Z)B\|^{1/(2q+1)} \leq (1 + \|\Omega_2\Omega_1^-\|)^{1/4q+2}\sigma_{k+1}$$

Based on above theorems, we can get error bound of truncated SVD.

Theorem 3. *Let A be an $m \times n$ matrix with singular values $\sigma_1 \geq \sigma_2 \cdots$, and let Z be an $m \times l$ matrix, where $l \geq k$. Suppose that $\hat{A}_{(k)}$ is a best rank-k approximation of $P_Z A$ with respect to the spectral norm then,*

$$\|A - \hat{A_{(k)}}\| \leq \sigma_{k+1} + \|(I - P_Z)A\|$$

1.2.2 Error bounds in the case of Gaussian test matrices

By drawing random Gaussian matrices we can get more concrete error bounds with average case and probability.

Theorem 4. *(Average case analysis) Suppose that $A, m \times n$ with singular values $\sigma_1 \geq \sigma_2 \cdots$ Choose a target rank $k \geq 2$ and an oversampling parameter $p \geq 2$, where $k + p \leq \min\{m, n\}$. Draw $n \times (k + p)$ standard Gaussian matrix Ω and construct the sample matrix $Y = A\Omega$ then,*

$$E\|(I - P_Y)A\|_F \leq (1 + \frac{k}{p-1})^{1/2}(\sum_{j>k}\sigma_j^2)^{1/2}$$

$$E\|(I - P_Y)A\| \leq (1 + \sqrt{\frac{k}{p-1}})\sigma_{k+1} + \frac{e\sqrt{k+p}}{p}(\sum_{j>k}\sigma_j^2)^{1/2}$$

Theorem 5. *(Probabilistic error bounds not about average) Under the frame of Theorem 4, assume further that $p \geq 4$ for all $u, t \geq 1$*

$$\|(I - P_Y)A\|_F \leq (1 + t\sqrt{\frac{12k}{p}})(\sum_{j>k}\sigma_j^2)^{1/2} + ut\frac{e\sqrt{k+p}}{p+1}\sigma_{k+1}$$

$$\|(I - P_Y)A\| \leq (1 + t\sqrt{\frac{12k}{p}})\sigma_{k+1} + t\frac{e\sqrt{k+p}}{p+1}(\sum_{j>k}\sigma_j^2)^{1/2} + ut\frac{e\sqrt{k+p}}{p+1}\sigma_{k+1}$$

with failure probability at most $5t^{-p} + 2e^{-u^2/2}$

1.2.3 Error bounds for SRFT test matrices

If we use SRFT instead of Gaussian matrices, result becomes as follows.

Theorem 6. *Fix an $m \times n$ matrix A with singular value $\sigma_1 \geq \sigma_2 \cdots$. Draw an $n \times l$ SRFT matrix Ω, where $4[\sqrt{k} + \sqrt{8log(kn)}]^2 log(k) \geq l \geq n$. Construct the sample matrix $Y = A\Omega$ then,*

$$\|(I - P_Y)A\| \leq \sqrt{1 + 7n/l}\sigma_{k+1}$$

$$\|(I - P_Y)A\|_F \leq \sqrt{1 + 7n/l}\sum_{j<k}(\sigma_j^2)^{1/2}$$

with failue probability at most $O(k^{-1})$

2 Discussions

2.1 Some questions

1. Some technical questions (hope to be helpful in the future)

   (a)
   $$\sup_{\|x\|^2+\|y\|^2} (\|\mathbf{A}\|^{1/2}\|x\| + \|\mathbf{B}\|^{1/2}\|y\|)^2 = \|\mathbf{A}\| + \|\mathbf{B}\|$$

   (b)
   $$[\sum_j d_j x_j^2]^t \leq [\sum_j d_j^t x_j^2]$$

2. $\|\|$ vs $\|\|_F$? which is used more often? any preference?
   my answ?: F is easy to extend to a tensor setting

3. How much better it is to do SVD for $\mathbf{Q}\mathbf{Q}^*\mathbf{A}$ instead of $\mathbf{A}$

4. It seems to me that a reason to use Gaussian test matrices and SRFT test matrices is we can measure error bounds with formula based on properties they already have. Can I get more random matrices something like that?

5. I can't understand fully a proof of SVD thm for higher order tensor. Can I get exercises or a paper which talks about n-mode product?

2.2 What results in the Tropp's paper can be (or have been) extended to higher-order tensors? How does the conclusion differ from the matrix case?

2.2.1 Change of overall setting

How can we change problem of seeking a matrix Q with k orthonormal columns? such that

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\| \leq \epsilon$$

and

$$\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^*\mathbf{A}\| \approx min_{rank(\mathbf{X})\leq k}\|\mathbf{A} - \mathbf{X}\|$$

We need to extend some definitions in tensor setting: Mainly tensor norm, tensor product and tensor SVD

Definition 1 (unfolding tensor). *Let* $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N}$, *then* $A_{(n)} \in \mathbb{C}^{I_nI_{n+1}\cdots I_NI_1\cdots I_{n-1}}$ *where* $a_{i_1,i_2,\cdots,i_N}$ *is at* $(I_n, (i_{n+1}-1)I_{n+2}\cdots I_NI_1 + (I_{n+2}-1)I_{n+3}\cdots I_nI_1\cdots I_{n-1} + \cdots + (I_N-1)I_1\cdots I_{n-1} + (i_1 - 1)I_2\cdots I_{n-1} + \cdots I_{n-1}$

Definition 2 (Rank1).

$$rank_n(\mathcal{A}) = rank(A_{(n)})$$

Definition 3 (Rank2).

$$R = rank(\mathcal{A}) = \arg\min_r\{r : \mathcal{A} = \sum_i^r U_i^{(1)} \circ \cdots \circ U_i^{(N)}\} \text{ where } U_i^{(j)} \in \mathbb{C}^{I_j}\}$$

Definition 4 (Inner Product). $\langle\mathcal{A},\mathcal{B}\rangle = \sum_{i_1}\cdots\sum_{i_N} b^*_{i_1,i_2\cdots,i_N}a_{i_1,i_2\cdots,i_N}$

Definition 5 (Orthogonality). *Arrays of which scalar product equals 0 are orthogoanl*

Definition 6 (Frobenius norm).

$$\|\mathcal{A}\| = \sqrt{\langle\mathcal{A},\mathcal{A}\rangle}$$

Definition 7 (Tensor Product). *Let* $\mathcal{A} \in \mathbb{C}^{I_1\times\cdots\times I_N}$ *and* $U \in \mathcal{C}^{J_n\times I_n}$. *Then,*

$$(\mathcal{A}\times_n U)_{i_1,\cdots,i_{n-1}j_ni_{n+1},\cdots,i_N} = \sum_{i_n} a_{i_1,\cdots,i_{n-1}i_ni_{n+1},\cdots,i_N}u_{j_ni_n}, \quad \mathcal{A}\times_n U \in \mathbb{C}^{I_1\times\cdots\times I_{n-1}J_nI_{n+1}\times\cdots\times I_N}$$

Now I get all definitions I need to generalize matrix version of SVD theorem into tensor version. First, I will write down matrix SVD with above notation.

Theorem 7 (Matrix SVD). *Every complex* $(I_1 \times I_2)$ *matrix* F *can be written as the product*

$$F = U^{(1)}SV^{(2)*} = S\times_1 U^{(1)}\times_2 V^{(2)*} = S\times_1 U^{(1)}\times_2 U^{(2)}$$

in which

1. $U^{(1)} = (U_1^{(1)}U_2^{(1)}\cdots U_{I_1}^{(1)})$ *is a unitary* $I_1\times I_1$ *matrix*
2. $U^{(2)} = (U_1^{(2)}U_2^{(2)}\cdots U_{I_2}^{(2)})$ *is a unitary* $I_2\times I_2$ *matrix*
3. S *is an* $I_1\times I_2$ *matrix with properties of*
   (a) *pseudodiagonality* $S = diag(\sigma_1,\sigma_2,\cdots,\sigma_{min(I_1,I_2)})$
   (b) *Ordering* $\sigma_1 \geq \sigma_2\cdots\sigma_{min(I_1,I_2)} \geq 0$

High order of SVD Theorem can be proved and as follows.

Theorem 8 (N-th order SVD). *Every complex* $(I_1\times\cdots\times I_N)$ *tensor* $\mathcal{A}$ *can be written as the product*

$$\mathcal{A} = S\times_1 U^{(1)}\times_2 U^{(2)}\cdots\times_N U^{(N)}$$

in which

1. *$U^{(n)} = (U_1^{(n)} U_2^{(n)} \cdots U_{I_n}^{(n)})$ is a unitary $I_n \times I_n$ matrix*
2. *S is an $(I_1 \times \cdots \times I_N)$ tensor of which the subtensors $S_{i_n=\alpha}$ obtained by fixing the nth index to α, have properties of*
   (a) *all-orthogonality: two subtensors $S_{i_n=\alpha}$ and $S_{i_n=\beta}$ are orthogonal for all possible values of n, α and β such that for $\alpha \neq \beta$, $\quad \langle S_{i_n=\alpha}, S_{i_n=\beta} \rangle = 0$*
   (b) *Ordering $\|S_{i_n=1}\| \geq \|S_{i_n=2}\| \geq \cdots \|S_{i_n=I_N}\| \geq 0$*

Based on this theorem, we can get the equivalent representations of singular value decomposition and this will give us a clue how to extend this paper results to higher order tensor:
If you unfold following tensor $\mathcal{A} = S \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}$ with regards to n, you can get

$$\mathcal{A}_{(n)} = U^{(n)} S_{(n)} \big(U^{(n+1)} \otimes U^{(n+2)} \otimes \cdots \otimes U^{(N)} \otimes U^{(1)} \otimes \cdots \otimes U^{(n-1)}\big)^t$$

Let's define $\Sigma^{(n)} = diag(\sigma_1^{(n)}, \sigma_2^{(n)}, \cdots, \sigma_{I_n}^{(n)}) \in \mathbb{R}^{I_n \times I_n}$ where $\sigma_i^{(n)}$ is Frobenius norm of each row from $S_{(n)}$. Also let $\tilde{S_{(n)}}$ is normalized matrix from $S_{(n)}$ with respect to $\Sigma^{(n)}$. Therefore, we can define

$$(V^{(n)})^H = \tilde{S_{(n)}} \big(U^{(n+1)} \otimes U^{(n+2)} \otimes \cdots \otimes U^{(N)} \otimes U^{(1)} \otimes \cdots \otimes U^{(n-1)}\big) \in \mathbb{C}^{I_{n+1} \cdots I_N I_1 \cdots I_{n-1} \times I_n}$$

Based on this, we can get a good relationship which helps us to compute higher order tensor SVD by getting matrix version of SVD.

$$A_{(n)} = U^{(n)} \Sigma^{(n)} (V^{(n)})^H$$

2.2.2 My first conjecture to generalize to higher order tensor

Exact SVD case: A key thing to find SVD for a given higher order tensor $\mathcal{A}$ is to unfolding a tensor and get $U^{(n)}$ for each unfolded matrix $A_{(n)}$.

Algorithm 1 Exact SVD

```
procedure SVD(A)
    for i ← 1N do
        Unfold A as A_(n)
        Find U^(n) in which A_(n) = U^(n) Σ^(n) (V^(n))^H using ordinary SVD for matrix case.
    Find S = A ×_1 U^(1) ×_2 U^(2) ··· ×_N U^(N)
    return (S, U^(1) ··· U^(n))
```

Approximate SVD case: My vague conjecture for this is that we can apply the paper's method to each unfolded matrix $A_{(n)}$. What I mean is that we can apply the randomized svd method for each step to find $U^{(n)}$in above algorithm.

Algorithm 2 Approx SVD

procedure SVD($\mathcal{A}$)
    for $i \leftarrow 1 : N$ **do**
        Unfold $\mathcal{A}$ as $A_{(n)}$
        Step A: Finding Q
        Generate an $I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N$ Gaussian test matrix Ω
        For $\mathbf{Y} = \mathbf{A_{(n)}}\Omega$
        Construct a matrix $\mathbf{Q}$ whose columns form an orthonormal basis for the range of $\mathbf{Y}$

        Step B: Finding $U^{(n)}$
        Form $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$
        Compute an SVD of the small matrix: $\mathbf{B} = \tilde{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^*$
        set $\mathbf{U^{(n)}} = Q\tilde{U}$
        This $U^{(n)}$ is a element of $A_{(n)} = U^{(n)}\Sigma^{(n)}(V^{(n)})^H$
    Find $\mathcal{S} = \mathcal{A} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}$
    return $(\mathcal{S}, U^{(1)} \cdots U^{(n)})$

Limitation: With this method, I couldn't find suitable upper error bound which was dealt with in the paper: Main hard point for apply this result is even though each $U^{(n)}$ can assure that

$$\|A_{(n)} - U^{(n)}\Sigma^{(n)}(V^{(n)})^H\| \leq \epsilon$$

It's hard to find a way to link to make following inequality hold.

$$\|\mathcal{A} - S \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}\| \leq \epsilon$$

I concluded that performing at svd each dimension only makes computation hard and can't guarantee upper error bound. My next method doesn't do SVD on each unfolded matrix but can find upper error bound successfully using previous results.

2.2.3 My new method to approximate SVD on higher order tensor

Instead of approximate SVD on each unfolded matrix like previous method, this method only find a matrix $\mathbf{Q}$ which can be done in Step A and use this each $\mathbf{Q}$ to approximate SVD. My new algorithm is as follows

Algorithm 3 Approx SVD 2

procedure SVD($\mathcal{A}$)
    Step A: Approximate SVD of $\mathcal{A}$
    for $i \leftarrow 1 : N$ **do**
        Unfold $\mathcal{A}$ as $A_{(n)}$
        Generate an $I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N$ Gaussian test matrix $\Omega^{(n)}$
        For $\mathbf{Y^{(n)}} = \mathbf{A_{(n)}}\Omega^{(n)}$
        Construct a matrix $\mathbf{Q}^{(n)}$ whose columns form an orthonormal basis for the range of $\mathbf{Y^{(n)}}$
        Form $\mathbf{P_{Y^{(n)}}} = \mathbf{Q^{(n)}}\mathbf{Q^{(n)*}}$
    get $\hat{\mathcal{A}} = \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_N P_{Y^{(N)}}$
    Step B: Get approximated SVD from $\hat{\mathcal{A}}$ **using Algorithm 1**
    return $(\mathcal{S}, U^{(1)} \cdots U^{(n)})$

By step A in Algorithm 3, we can a get good approximation $\hat{\mathcal{A}}$ for $\mathcal{A}$ such that $\|\mathcal{A} - \hat{\mathcal{A}}\| \leq \epsilon$. In Step B, we are using exact SVD Algorithm 1 for $\hat{\mathcal{A}}$. Therefore, we can get a approximated SVD of higher order tensor through Algorithm 3.
The only thing left is to prove that we have an upper bound error for $\|\mathcal{A} - \hat{\mathcal{A}}\|$, in which

$$\hat{\mathcal{A}} = \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_N P_{Y^{(N)}}$$

Theorem 9. *(Probabilistic error bounds for higher order tensor case) Under the frame of Theorem 4, assume further that for each n,* $k_n \geq 2 \quad p_n \geq 4$, $u, t \geq 1$ *and let* $A_{(n)}$ *be unfolded matrix of a given tensor* $\mathcal{A}$

$$\|\mathcal{A} - \hat{\mathcal{A}}\| \leq \sum_{n=1}^{N} \left((1 + t\sqrt{\frac{12k_n}{p_n}})(\sum_{j>k} \sigma_j^2)^{1/2} + ut\frac{e\sqrt{k_n + p_n}}{p_n + 1}\sigma_{k_n+1} \right)$$

with failure probability at most $1 - \prod_{n=1}^{N}(1 - 5t^{-p_n} + 2e^{-u^2/2})$

Proof: Let $\hat{\mathcal{E}_N} = \mathcal{A} - \hat{\mathcal{A}}$

$$\begin{aligned}
\|\hat{\mathcal{E}_N}\|_F &= \|\mathcal{A} - \mathcal{A} \times_N P_{Y^{(N)}} + \mathcal{A} \times_N P_{Y^{(N)}} - \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_N P_{Y^{(N)}}\|_F \\
&\leq \|\mathcal{A} - \mathcal{A} \times_N P_{Y^{(N)}}\|_F + \|\mathcal{A} \times_N P_{Y^{(N)}} - \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_N P_{Y^{(N)}}\|_F \\
&= \|\mathcal{A} - \mathcal{A} \times_N P_{Y^{(N)}}\|_F + \|(\mathcal{A} - \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_{N-1} P_{Y^{(N-1)}}) \times_N P_{Y^{(N)}}\|_F \\
&\leq \|\mathcal{A} - \mathcal{A} \times_N P_{Y^{(N)}}\|_F + \|\mathcal{A} - \mathcal{A} \times_1 P_{Y^{(1)}} \times_2 P_{Y^{(2)}} \cdots \times_{N-1} P_{Y^{(N-1)}}\| \\
&= \|\mathcal{A} - \mathcal{A} \times_N P_{Y^{(N)}}\|_F + \|\hat{\mathcal{E}_{N-1}}\|_F \\
\cdots &\leq \sum_{n=1}^{N} \|\mathcal{A} - \mathcal{A} \times_n P_{Y^{(n)}}\|_F = \sum_{n=1}^{N} \|(I - P_{Y^{(n)}})A_{(n)}\|_F
\end{aligned}$$

by the Theorem 5, we can get

$$\|(I - P_{Y^{(n)}})A_{(n)}\|_F \leq (1 + t\sqrt{\frac{12k_n}{p_n}})(\sum_{j>k} \sigma_j^2)^{1/2} + ut\frac{e\sqrt{k_n + p_n}}{p_n + 1}\sigma_{k_n+1}$$

with failure probability at most $5t^{-p_n} + 2e^{-u^2/2}$
Therefore, by plugging this inequality into above we can get the result. □

2.3 Further things to do

There are a few things I want to deal with for next week

1. I read some main parts of a paper: A MULTILINEAR SINGULAR VALUE DECOMPOSITION. I want to finish reading this paper to have in-depth understanding about tensor svd and will read COMPUTATIONAL AND STATISTICAL BOUNDARIES FOR SUBMATRIX LOCALIZATION IN A LARGE NOISY MATRIX.

2. I couldn't think about What are the current state-of-art, open questions, and challenges? and application things.
   I want to search and think more for next week

3. How can I choose suitable p and k on each unfolded matrix is another thing to discuss.

4. Stat 771 project : I want to start do research project and want to study what would be helpful for my future research. I am not sure detailed topic, I want to find subject during this week(I want to do some projects more applicable so it will be on the same line of finding possible applications of my results)