Comparisons of two tensor SVD algorithms and Ordinal tensor modeling

Chanwoo Lee

1 Comparison of 2 different algorithms for SVD

1.1 Simulation procedure

Two methods for SVD tensor are as follows.

Algorithm 1 Approx tensor SVD 1

procedure SVD($\mathcal{A}$)
  Step A: Approximate SVD of $\mathcal{A}$
  for $n \leftarrow 1 : N$ **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 $\mathbf{P}_{\mathbf{Y^{(n)}}} = \mathbf{Q^{(n)}} \mathbf{Q^{(n)*}}$
  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
  $\mathcal{S} = \mathcal{A} \times_1 Q^{(1)*} \times_2 Q^{(2)*} \cdots \times_N Q^{(N)*}$
  for $i \leftarrow 1 : N$ **do**
    $U^{(i)} = Q^{(i)}$
  return $(\mathcal{S}, U^{(1)} \cdots U^{(n)})$

Algorithm 2 Approx tensor SVD 2

procedure SVD($\mathcal{A}$)
  Step A: Approximate SVD of $\mathcal{A}$
  for $i \leftarrow 1 : N$ **do**
    Get Gaussian test matrix Ω_n whose size is $I_i \times (k_i + p)$
    Form $A^{(i)} = \text{Unfold}_i(\mathcal{A} \times_1 \Omega_1^* \times \cdots \times_{i-1} \Omega_{i-1}^* \times_{i+1} \Omega_{i+1}^* \times \cdots \times_N \Omega_N^*)$
    Find a matrix $Q^{(i)}$ whose size is $I_i \times (k_i + p)$
    $U^{(i)} = Q^{(i)}$
  get $\mathcal{S} = \mathcal{A} \times_1 Q^{(1)*} \times_2 Q^{(2)*} \cdots \times_N Q^{(N)*}$
  return $(\mathcal{S}, U^{(1)} \cdots U^{(n)})$

Comparison simulation procedure is as follows.

1. Make $\boldsymbol{a}, \boldsymbol{b}$ and $\boldsymbol{c}$, 100×1 matrices drawn from $N(0, 1)$
2. Construct a signal $\mathcal{X}$, $100 \times 100 \times 100$ tensor having rank of 1 s.t. $\mathcal{X} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}$
3. Make a noise $\mathcal{E}$, $100 \times 100 \times 100$ tensor drawn from $N(0, \sigma^2)$
4. Get a $\mathcal{D} = \mathcal{X} + \mathcal{E}$ and estimate $\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{c}}$ using above 2 methods.
5. Check MSE ($\|\mathcal{D} - \hat{\boldsymbol{a}} \circ \hat{\boldsymbol{b}} \circ \hat{\boldsymbol{c}}\|_F^2$), angle between $\boldsymbol{a}$ vs $\hat{\boldsymbol{a}}$, $\boldsymbol{b}$ vs $\hat{\boldsymbol{b}}$ and $\boldsymbol{c}$ vs $\hat{\boldsymbol{c}}$
6. Repeat 1000 times and average them for each σ from $0.05, 0.1, \cdots 1.0$.

1.2 Simulation result

Simulation result is as follows

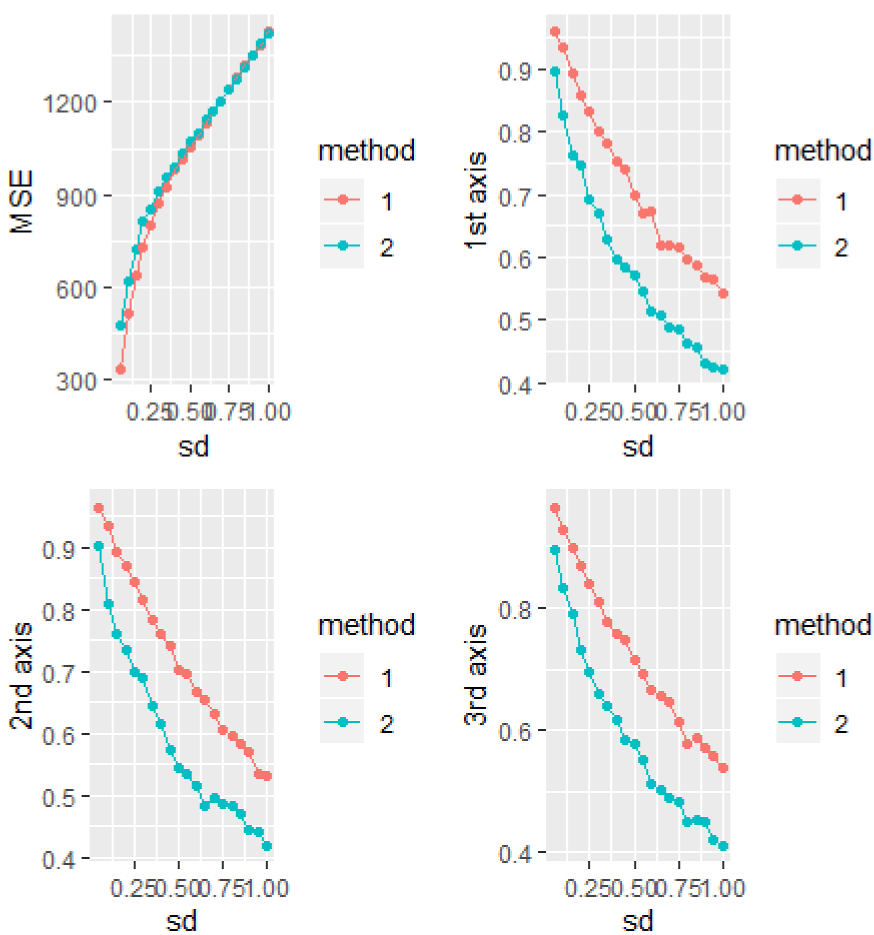

Figure 1: Accuracy of two methods according to different noise standard deviation

1.3 My explanation for this phenomenon

Our goal is to estimate $\mathcal{X} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}$ from $\mathcal{D} = \mathcal{X} + \mathcal{E}$ where $\mathcal{E} \sim N(0,1)$ and $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are 100×1 matrices whose elements are from $N(0,1)$. However $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ such that $\mathcal{X} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}$ are not unique because we can make them different by constant multiplication, we will measure accuracy of our estimate by angle difference between $\boldsymbol{a}, \hat{\boldsymbol{a}}$, $\boldsymbol{b}, \hat{\boldsymbol{b}}$ and $\boldsymbol{c}, \hat{\boldsymbol{c}}$
For each method, let's try to find angle of $\boldsymbol{a}, \hat{\boldsymbol{a}}$. Other axes would be the same. We find estimate $\hat{\boldsymbol{a}}$ as follows.

1. 1st method.

   $\mathcal{D} = \mathcal{X} + \mathcal{E},$ Unfold $\mathcal{D}$ into 1st axis , generate random matrix $\Omega_{(1)}$ and find $\hat{\boldsymbol{a}}$ such that
   $$\underbrace{\mathcal{D}_{(1)}}_{d \times d^2} \underbrace{\Omega_{(1)}}_{d^2 \times 1} = (\mathcal{X}_{(1)} + \mathcal{E}_{(1)})\Omega_{(1)} = \underbrace{\mathcal{X}_{(1)}\Omega_{(1)}}_{d \times 1} + \underbrace{\mathcal{E}_{(1)}\Omega_{(1)}}_{d \times 1} = \underbrace{\hat{\boldsymbol{a}}}_{d \times 1} \underbrace{R}_{1 \times 1}$$
   therefore $\hat{\boldsymbol{a}}$ is parallel to $\mathcal{X}_{(1)}\Omega_{(1)} + \mathcal{E}_{(1)}\Omega_{(1)}$ and $\boldsymbol{a}$ is parallel to $\mathcal{X}_{(1)}\Omega_{(1)}(\because \mathcal{X}_{(1)} = \boldsymbol{a}(\boldsymbol{b} \otimes c)^T)$
   Finally we can get a following, $\cos\Theta(\boldsymbol{a}, \hat{\boldsymbol{a}}) = \dfrac{\langle \mathcal{X}_{(1)}\Omega_{(1)}, \mathcal{X}_{(1)}\Omega_{(1)} + \mathcal{E}_{(1)}\Omega_{(1)} \rangle}{\|\mathcal{X}_{(1)}\Omega_{(1)}\| \|\mathcal{X}_{(1)}\Omega_{(1)} + \mathcal{E}_{(1)}\Omega_{(1)}\|}$
   Note that each element of $\mathcal{E}_{(1)}\Omega_{(1)}$ follows $N(0,1)^2 \equiv \chi^2(1)$

2. 2nd method.

   $\mathcal{D} = \mathcal{X} + \mathcal{E},$ Unfold $\mathcal{D}$ into 1st axis , generate random matrix $\Omega_{(2)}, \Omega_{(3)}$ and $\tilde{\Omega}_{(1)}$ and find $\hat{\boldsymbol{a}}$ such that
   $$\underbrace{[\underbrace{D}_{d \times d \times d} \times_2 \underbrace{\Omega_{(2)}}_{d \times 1} \times_3 \underbrace{\Omega_{(3)}}_{d \times 1}]_{(1)}}_{d \times 1} \underbrace{\tilde{\Omega}_{(1)}}_{1 \times 1} = \underbrace{\hat{\boldsymbol{a}}}_{d \times 1} \underbrace{R}_{1 \times 1}$$
   If we elaborate more we can get following expression
   $$[\mathcal{D} \times_2 \Omega_{(2)} \times_3 \Omega_{(3)}]_{(1)}\tilde{\Omega}_{(1)} = [(\mathcal{X} + \mathcal{E}) \times_2 \Omega_{(2)} \times_3 \Omega_{(3)}]_{(1)}\tilde{\Omega}_{(1)} = \underbrace{(\mathcal{X} + \mathcal{E})_{(1)}}_{d \times d^2} \underbrace{(\Omega_{(3)} \otimes \Omega_{(2)})^T}_{d^2 \times 1} \underbrace{\tilde{\Omega}_{(1)}}_{1 \times 1}$$
   $$= \underbrace{\mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}}_{d \times 1} + \underbrace{\mathcal{E}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}}_{d \times 1} = \underbrace{\hat{\boldsymbol{a}}}_{d \times 1} \underbrace{R}_{1 \times 1}$$
   therefore $\hat{\boldsymbol{a}}$ is parallel to $\mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)} + \mathcal{E}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}$
   and $\boldsymbol{a}$ is parallel to $\mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)} \quad (\because \mathcal{X}_{(1)} = \boldsymbol{a}(\boldsymbol{b} \otimes c)^T)$
   Finally we can get a following,
   $$\cos\tilde{\Theta}(\boldsymbol{a}, \hat{\boldsymbol{a}}) = \frac{\langle X_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}, \mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)} + \mathcal{E}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)} \rangle}{\|\mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}\| \|\mathcal{X}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)} + \mathcal{E}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}\|}$$
   Note that each element of $\mathcal{E}_{(1)}(\Omega_{(3)} \otimes \Omega_{(2)})^T\tilde{\Omega}_{(1)}$ follows $N(0,1)^4$

To sum up, angle between true and estimated $\boldsymbol{a}$ is

$\cos\Theta(\boldsymbol{a}, \hat{\boldsymbol{a}}) = \dfrac{\langle \mathcal{X}_{(1)}\Omega_{(1)}, \mathcal{X}_{(1)}\Omega_{(1)} + \mathcal{E}_{(1)}\Omega_{(1)} \rangle}{\|\mathcal{X}_{(1)}\Omega_{(1)}\| \|\mathcal{X}_{(1)}\Omega_{(1)} + \mathcal{E}_{(1)}\Omega_{(1)}\|}$ for 1st method. 2nd method is as follows

$$\cos\tilde{\Theta}(\boldsymbol{a},\hat{\boldsymbol{a}}) = \frac{\langle X_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)}, \mathcal{X}_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)} + \mathcal{E}_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)}\rangle}{\|\mathcal{X}_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)}\|\|\mathcal{X}_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)} + \mathcal{E}_{(1)}(\Omega_{(3)}\otimes\Omega_{(2)})^T\tilde{\Omega}_{(1)}\|}$$

So I did further simulations to find a mean of each $\cos\theta$'s using Monte carlo simulation and it turn out that for given $\mathcal{X}$ and $\mathcal{E}$,

$$E(\cos\Theta) = 0.5153142 \qquad E(\cos\tilde{\Theta}) = 0.3865333$$

This coincides with the above simulation result.

2 Ordinal tensor model proposal

1. For the category tensor $\mathcal{Y} = [y_{i_1,\cdots,i_N}] \in \{1,2,\cdots,K\}^{d_1\times\cdots\times d_N}$, I assume it's entries are realization of multinomial random variables. My model for those kinds of tensors is as follows.

   $$P(y_{i_1,\cdots,i_N} = c) = [\mathbf{f}_\beta(\theta_{i_1,\cdots,i_N})]_c \quad c\in\{1,2,\cdots,K\} \quad \beta\in\mathcal{R}^{K-1}$$

   where $\mathbf{f}_\beta$ is a link function such that.

   $$\mathbf{f}_\beta(x) = \left[\frac{e^{\beta_1^* x}}{\sum_{t=1}^K e^{\beta_t^* x}},\cdots,\frac{e^{\beta_K^* x}}{\sum_{t=1}^K e^{\beta_t^* x}}\right]^T = \left[\frac{e^{\beta_1 x}}{1+\sum_{t=1}^{K-1} e^{\beta_t x}},\cdots,\frac{e^{\beta_{K-1} x}}{1+\sum_{t=1}^{K} e^{\beta_t x}},\frac{1}{1+\sum_{t=1}^{K} e^{\beta_t x}}\right]^T$$

   Also, we assume the parameter tensor $\Theta = \mathcal{C}\times_1 A_1\times_2 A_2\cdots\times_N A_N$ where $\mathcal{C}\in\mathcal{R}^{r_1\times\cdots\times r_N}$ and $A_i\in\mathcal{R}^{d_i\times r_i}$ To sum up we have the following model.

   $$\mathcal{Y} = \arg\max_c [\mathbf{f}_\beta(\Theta)]_c$$

2. For the ordinary tensor $Y = [y_{i_1,\cdots,i_N}] \in \{1,2,\cdots,K\}^{d_1\times\cdots\times d_N}$ we will use cumulative logit model. Our model is as follows.

   $$P(y_{i_1,\cdots,i_N} \le j|x) = \pi_1(x)+\cdots+\pi_j(x)$$

   $$logit(P(y_{i_1,\cdots,i_N} \le j|x)) = \log\frac{P(y_{i_1,\cdots,i_N} < j|x)}{1-P(y_{i_1,\cdots,i_N} \le j|x)} = \log\frac{\pi_1(x)+\cdots+\pi_j(x)}{\pi_{j+1}+\cdots+\pi_K(x)} = \alpha_j+\beta x$$

   where α_j is non-decreasing with regards to j

   $$P(y_{i_1,\cdots,i_N} = j) = [\mathbf{f}_{\alpha\beta}(\theta_{i_1,\cdots,i_N})]_j \quad j\in\{1,2,\cdots,K\} \quad \alpha\in\mathcal{R}^K \quad \beta\in\mathcal{R}$$

   where

   $$\mathbf{f}_{\alpha,\beta}(x) = \left[\frac{e^{\alpha_1+\beta x}}{1+e^{\alpha_1+\beta x}},\frac{e^{\alpha_2+\beta x}}{1+e^{\alpha_2+\beta x}}-\frac{e^{\alpha_1+\beta x}}{1+e^{\alpha_1+\beta x}},\cdots,\frac{e^{\alpha_K+\beta x}}{1+e^{\alpha_K+\beta x}}-\frac{e^{\alpha_{K-1}+\beta x}}{1+e^{\alpha_{K-1}+\beta x}},\right]^T$$

   Like in the category tensor case, we assume the parameter tensor $\Theta = \mathcal{C}\times_1 A_1\times_2 A_2\cdots\times_N A_N$ where $\mathcal{C}\in\mathcal{R}^{r_1\times\cdots\times r_N}$ and $A_i\in\mathcal{R}^{d_i\times r_i}$ Finally, we have the following model.

   $$\mathcal{Y} = \arg\max_j [\mathbf{f}_{\alpha,\beta}(\Theta)]_j$$

3. Another model for the ordinary tensor $Y = [y_{i_1,\cdots,i_N}] \in \{1, 2, \cdots, K\}^{d_1\times\cdots\times d_N}$ is using threshold on latent tensor $\Theta = \mathcal{C} \times_1 A_1 \times_2 A_2 \cdots \times_N A_N$
where $\mathcal{C} \in \mathcal{R}^{r_1\times\cdots\times r_N}$ and $A_i \in \mathcal{R}^{d_i\times r_i}$. We put extra threshold parameter $t_1, t_2, \cdots, t_{K-1}$ with $t_0 = -\infty, \quad t_K = \infty$ and assign response variable as

$$y_{i_1,\cdots,i_N} = j \quad \text{ if } \theta_{i_1,\cdots,i_N} \in [t_{j-1}, t_j)$$

3 Problem: matrix approximation can be extended to tensor approximation

Theorem 1. *Let $\mathcal{A} = C \times_1 M_1 \times_2 M_2 \times_3 M_3 \in \mathcal{R}^{d_1\times d_2\times d_3}$ where $C \in \mathcal{R}^{r_1\times r_2\times r_3}$ and $M_i \in \mathcal{R}^{d_i\times r_i}$ for each i,*
Suppose we have estimation $\hat{M}_1, \hat{M}_2, \hat{M}_3$ such that $\|M_i - \hat{M}_i\| \leq \epsilon$ for each i Let $\hat{C} = \mathcal{A} \times_1 \hat{M}_1^{\ t} \times_2 \hat{M}_2^{\ t} \times_3 \hat{M}_3^{\ t}$, and $\hat{A} = \hat{C} \times_1 \hat{M}_1 \times_2 \hat{M}_2 \times_3 \hat{M}_3$
Then what can you get for error abound for $\|\hat{A} - \mathcal{A}\|$?

Proof. First, notice that for each i,

$$\begin{aligned}\|M_iM_i^t - \hat{M}_i\hat{M}_i^{\ t}\| = \|M_iM_i^t - M_i\hat{M}_i^{\ t} + M_i\hat{M}_i^{\ t} - \hat{M}_i\hat{M}_i^{\ t}\| &= \|M_i(M_i^t - \hat{M}_i^{\ t}) + (M_i - \hat{M}_i)\hat{M}_i^{\ t}\| \\ &\leq (2\|M_i\| + \epsilon)\epsilon\end{aligned}$$

Main proof is as follows

$$\begin{aligned}
\|\mathcal{A} - \hat{A}\| &= \|\mathcal{A} - \hat{C} \times_1 \hat{M}_1 \times_2 \hat{M}_2 \times_3 \hat{M}_3\| = \|\mathcal{A} - \mathcal{A} \times_1 \hat{M}_1^{\ t} \times_2 \hat{M}_2^{\ t} \times_3 \hat{M}_3^{\ t} \times_1 \hat{M}_1 \times_2 \hat{M}_2 \times_3 \hat{M}_3\| \\
&= \|\mathcal{A} - \mathcal{A} \times_1 \hat{M}_1\hat{M}_1^{\ t} \times_2 \hat{M}_2\hat{M}_2^{\ t} \times_2 \hat{M}_2\hat{M}_2^{\ t}\| \\
&= \|A_{(1)} - \hat{M}_1\hat{M}_1^{\ t}A_{(1)}(\hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t})\| \\
&= \|M_1M_1^tA_{(1)}(M_2M_2^t \otimes M_3M_3^t) - \hat{M}_1\hat{M}_1^{\ t}A_{(1)}(\hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t})\| \\
&= \|(M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t})A_{(1)}(M_2M_2^t \otimes M_3M_3^t) + \hat{M}_1\hat{M}_1^{\ t}A_{(1)}(M_2M_2^t \otimes M_3M_3^t - \hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t})\| \\
&= \|(M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t})A_{(1)}(M_2M_2^t \otimes M_3M_3^t)\| \\
&\quad + \|\hat{M}_1\hat{M}_1^{\ t}A_{(1)}(M_2M_2^t \otimes M_3M_3^t - \hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t})\| \\
&\leq \|M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t}\|\|A_{(1)}\|\|M_2M_2^t \otimes M_3M_3^t\| \\
&\quad + \|\hat{M}_1\hat{M}_1^{\ t}\|\|A_{(1)}\|\|M_2M_2^t \otimes M_3M_3^t - \hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t}\| \\
&\leq (\|M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t}\| + \|M_2M_2^t \otimes M_3M_3^t - \hat{M}_2\hat{M}_2^{\ t} \otimes \hat{M}_3\hat{M}_3^{\ t}\|)\|\mathcal{A}\| \\
&\leq (\|M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t}\| + \|(M_2M_2^t - \hat{M}_2\hat{M}_2^{\ t}) \otimes M_3M_3^t\| + \|\hat{M}_2\hat{M}_2^{\ t} \otimes (M_3M_3^t - \hat{M}_3\hat{M}_3^{\ t})\|)\|\mathcal{A}\| \\
&\leq (\|M_1M_1^t - \hat{M}_1\hat{M}_1^{\ t}\| + \|M_2M_2^t - \hat{M}_2\hat{M}_2^{\ t}\| + \|M_3M_3^t - \hat{M}_3\hat{M}_3^{\ t}\|)\|\mathcal{A}\| \\
&\leq \|\mathcal{A}\|(2\|M_1\| + 2\|M_2\| + 2\|M_3\| + 3\epsilon)\epsilon
\end{aligned}$$

□

To get better error bound let's define principal angles.

Definition 1. *For nonzero subspaces $\mathcal{R}, \mathcal{N} \subset \mathbb{R}^n$, the minimal angle between $\mathcal{R}$ and $\mathcal{N}$ is defined to be the number $0 \leq \theta \leq \pi/2$ that satisfies*

$$\cos\theta = \max_{u \in \mathcal{R}, v \in \mathcal{N} \|u\| = \|v\| = 1} v^t u.$$

Then, our new error bound becomes as in following Theorem 2.

Theorem 2. *Under the same condition in Theorem 1 but* $\sin(\theta(span(M_i), span(\hat{M}_i^t))) < \epsilon$ *with matrix norm,*

$$\|\mathcal{A} - \hat{A}\| \leq 6\epsilon\|\mathcal{A}\|$$

Proof. It suffices to show $\|M_i M_i^t - \hat{M}_i \hat{M}_i^t\| \leq 2\epsilon$ because we can apply this last inequality in the proof of Theorem 1.

$$\|\mathcal{A} - \hat{A}\| \leq (\|M_1 M_1^t - \hat{M}_1 \hat{M}_1{}^t\| + \|M_2 M_2^t - \hat{M}_2 \hat{M}_2{}^t\| + \|M_3 M_3^t - \hat{M}_3 \hat{M}_3{}^t\|)\|\mathcal{A}\| \leq 6\epsilon\|\mathcal{A}\|$$

Then we are done.
Proof of the above inequality is as follows.

$$\begin{aligned}
\|M_i M_i^t - \hat{M}_i \hat{M}_i^t\| &= \|M_i M_i^t - \hat{M}_i \hat{M}_i^t\| \\
&= \|M_i M_i^t - M_i M_i^t \hat{M}_i \hat{M}_i^t + M_i M_i^t \hat{M}_i \hat{M}_i^t - \hat{M}_i \hat{M}_i^t a\| \\
&\leq \|M_i M_i^t - M_i M_i^t \hat{M}_i \hat{M}_i^t\| + \|M_i M_i^t \hat{M}_i \hat{M}_i^t - \hat{M}_i \hat{M}_i^t\| \\
&= \|M_i M_i^t (I - \hat{M}_i \hat{M}_i^t)\| + \|(M_i M_i^t - I)\hat{M}_i \hat{M}_i^t\| \\
&\leq \sin(\theta) + \sin(\theta) \leq 2\epsilon
\end{aligned}$$

Last inequality follows from combining following 2 lemmas. □

Lemma 1. *If $P_R$and P_N are the orthogonal projectors onto $R$and N , respectively, then*

$$\cos\theta = \|P_N P_R\| = \|P_R P_N\|.$$

Proof. For vectors x and y such that $\|x\| = \|y\| = 1$, we have $P_R x \in \mathcal{R}$ and $P_N y \in \mathcal{N}$ Then

$$\cos\theta = \max_{u \in \mathcal{R}, v \in \mathcal{N} \|u\| = \|v\| = 1} v^t u = \max_{u \in \mathcal{R}, v \in \mathcal{N}, \|u\| \leq 1 \|v\| \leq 1} v^t u = \max_{\|x\| \leq 1 \|y\| \leq 1} y^t P_{\mathcal{N}} P_{\mathcal{R}} x = \|P_R P_N\|$$

.

□

Lemma 2. *Under the same condition on Lemma 1,*

$$\|P_N(I - P_R)\| \leq \sin(\theta)$$

Proof.

$$\begin{aligned}
\|P_N(I - P_R)\|^2 &= \max_{u \in \mathcal{R}, \|u\| = 1} u^t P_N (I - P_R) u = \max_{u \in \mathcal{R}, \|u\| = 1} u^t P_N u - u^t P_N P_R u \\
&\leq 1 - \|P_N P_R\|^2 = \sin^2(\theta)
\end{aligned}$$

□

4 Question.

Appendix

A Comparison simulation

```
## tensor_svd approx first method.
tensor_svd = function(tnsr,k1,k2,k3,p){
  App = list(Z=NULL,U=NULL)
  mat1 <- k_unfold(tnsr,m=1)
  mat2 <- k_unfold(tnsr,m=2)
  mat3 <- k_unfold(tnsr,m=3)
  Q1 <- StepAm(mat1@data,k1,p)
  Q2 <- StepAm(mat2@data,k2,p)
  Q3 <- StepAm(mat3@data,k3,p)
  Coreten <- ttm(ttm(ttm(tnsr,t(Q1),1),t(Q2),2),t(Q3),3)
  App$Z = Coreten
  App$U = list(Q1,Q2,Q3)
  return(App)
}


tensor_svd2 = function(tnsr,k1,k2,k3,p){
  App = list(Z=NULL,U=NULL)
  rk = c(k1,k2,k3)
  a = c(1,2,3,1,2,3)
  Omega = list()
  Q = list()
  for (i in 1:3) {

    Omega[[i]] <- matrix(rnorm(tnsr@modes[i]*(rk[i]+p)),ncol = rk[i]+p)
  }
  for (i in 1:3) {
    ing <- matrix(rnorm(prod(rk+p)),ncol = rk[i]+p)
    tmp <- k_unfold(ttm(ttm(tnsr,t(Omega[[a[i+1]]]),a[i+1]),t(Omega[[a[i
    +2]]]),a[i+2]),m =i)@data%*%ing
    Q[[i]] <- qr.Q(qr(tmp))
  }
  Coreten <- ttm(ttm(ttm(tnsr,t(Q[[1]]),1),t(Q[[2]]),2),t(Q[[3]]),3)
  App$Z <- Coreten
  App$U <- Q
  return(App)
}



tensor_resid = function(tnsr,App){
  a = normf(tnsr-ttm(ttm(ttm(App$Z,App$U[[1]],1),App$U[[2]],2),App$U
   [[3]],3))
  return(a)
}

```

```
angle = function(u,t){
  return(inner(u,t)/sqrt(inner(u,u)*inner(t,t)))
}
set.seed(20)
a <- as.matrix(rnorm(100))
b <- as.matrix(rnorm(100))
c <- as.matrix(rnorm(100))
tnsr <- as.tensor(array(1,dim = c(1,1,1)))
X <- ttm(ttm(ttm(tnsr,a,1),b,2),c,3)

result = as.data.frame(matrix( nrow = 2000, ncol = 5))
names(result) <- c("method","MSE","1st angle","2nd angle","3rd angle")
for (i in 1:1000) {
  set.seed(1000*index+i)
  eps <- as.tensor(array(rnorm(1000000,mean =0,sd = index/20),dim = c
    (100,100,100)))
  result[i,1] <- "1st method"
  result[i,2] <- tensor_resid(X+eps,tensor_svd(X+eps,1,1,1,0))
  result[i,3] <- angle(tensor_svd(X+eps,1,1,1,0)$U[[1]],a)
  result[i,4] <- angle(tensor_svd(X+eps,1,1,1,0)$U[[2]],b)
  result[i,5] <- angle(tensor_svd(X+eps,1,1,1,0)$U[[3]],c)
  result[i+1000,1] <- "2nd method"
  result[i+1000,2] <- tensor_resid(X+eps,tensor_svd2(X+eps,1,1,1,0))
  result[i+1000,3] <- angle(tensor_svd2(X+eps,1,1,1,0)$U[[1]],a)
  result[i+1000,4] <- angle(tensor_svd2(X+eps,1,1,1,0)$U[[2]],b)
  result[i+1000,5] <- angle(tensor_svd2(X+eps,1,1,1,0)$U[[3]],c)
}


write.csv(result,paste("result3_",index,".csv",sep=""),row.names=FALSE)
```

B Monte carlo simulation to find out mean of $\cos\theta$

```
a <- as.matrix(rnorm(100))
b <- as.matrix(rnorm(100))
c <- as.matrix(rnorm(100))
tnsr <- as.tensor(array(1,dim = c(1,1,1)))
X <- ttm(ttm(ttm(tnsr,a,1),b,2),c,3)

X_1 = k_unfold(X,1)@data
eps <- k_unfold(as.tensor(array(rnorm(1000000),dim = c(100,100,100))),1)
    @data


x <- matrix(rnorm(10000),nrow=10000)
omeg_1 <- matrix(rnorm(100),nrow=100)
omeg_2 <- matrix(rnorm(100),nrow=100)
omg <- rnorm(1)
y <- omeg_1%x%omeg_2*omg
angl = matrix(nrow = 10000, ncol = 2)
names(angl) <- c("1st","2nd")
  for (i in 1:10000) {
    set.seed(i)
```

```
    x <- matrix(rnorm(10000),nrow=10000)
    omeg_1 <- matrix(rnorm(100),nrow=100)
    omeg_2 <- matrix(rnorm(100),nrow=100)
    omg <- rnorm(1)
    y <- omeg_1%x%omeg_2*omg
    angl[i,1] <- angle(X_1%*%x,(X_1+eps)%*%x)
    angl[i,2] <- angle(X_1%*%y,(X_1+eps)%*%y)
  }
apply(angl,2,mean)
```