On the Implementation of Tucker Decomposition

Yu-Hsuan Huang
asd00012334@hotmail.com

November 2019

Abstract

In this assignment we implemented Tucker's decomposition and a relaxed variant. Our implementation is based on a number of iterations. We first introduce Tucker decomposition and its variant. Then elaborate key ideas of iterations including initialization. Finally, we present some benchmarks and give several comments on the experiment results.

1 Introduction

The Tucker decomposition is just generalization of SVD in the sense that they both require orthogonality.

Definition 1 (Tucker Decomposition). *Given any collection of finite index sets $\mathcal{J}_1, \ldots, \mathcal{J}_k < \mathbb{N}$ with $\mathcal{J} := \mathcal{J}_1 \times \cdots \times \mathcal{J}_k$ and $T \in \mathbb{C}^{\mathcal{J}}$, a Tucker decomposition consists of a family of orthonormal subsets*

$$\{U_{\ell i}\}_{1 \leq i \leq r_\ell} \subseteq \mathbb{C}^{\mathcal{J}_\ell}, \text{ with } \langle U_{\ell i_1}, U_{\ell i_2} \rangle = \delta_{i_1 i_2},$$

and a core tensor $g \in \mathbb{C}^{[r_1] \times \cdots \times [r_k]}$, where for $j_\ell \in \mathcal{J}_\ell$

$$T_{j_1 \ldots j_k} = \sum_{i_1, \ldots, i_k \in [r_1] \times \cdots \times [r_k]} g_{i_1 \ldots i_k} \prod_{1 \leq \ell \leq k} U_{\ell, i_\ell, j_\ell}.$$

When equality is relaxed to approximation, we called this Tucker approximation of T.

For convenience, there are two ways to rewrite this. First, in regular tensor notation, one could write,

$$T = \sum_{i_1, \ldots, i_k \in [r_1] \times \cdots \times [r_k]} g_{i_1 \ldots i_k} \bigotimes_{1 \leq \ell \leq k} U_{\ell i_\ell}.$$

This discard the detail information of indices. When specified each indices, we could have the so called *Einstein's summation convention* as following,

$$T_{j_1 \ldots j_k} = g_{i_1 \ldots i_k} \prod_{1 \leq \ell \leq k} U_{\ell, i_\ell, j_\ell}.$$

Here, we splits indices into two kinds, those appears on both sides of equation, j_ℓ, called the free indices, and those appears only at one side, i_ℓ, called the dummy indices. The rule is we should sum over of all dummy indices and not the free indices. In our later disscussion, wihtout further specifying, we will use these two notations.

2 Implementaion Procedures

For the problem set, we first consider a relaxed Tucker decomposition.

Exercise 1. *Set $k = 3$, indices j_ℓ, i_ℓ running in $\mathcal{J}_\ell, [r_\ell]$. For $T \in \mathbb{C}^{\mathcal{J}}$, find tensor approximation $A = U_1, B = U_2, C = U_3$,*

$$T_{j_1 j_2 j_3} \approx g_{i_1 i_2 i_3} A_{i_1 j_1} B_{i_2 j_2} C_{i_3 j_3},$$

except that now A, B, C need not be orthogonal. Instead, restrict $\|A_{i_1}\| = \|B_{i_2}\| = \|C_{i_3}\| = 1$.

From basic linear algebra, any second degree tensor, a matrix, admits its pseudo inverse from the SVD form.

Lemma 1 (Pseudo Inverse)**.** *Given a matrix $A \in \mathbb{C}^{\mathcal{J}_R \times \mathcal{J}_C}$ with SVD,*

$$A_{ij} = \sigma_k u_{ik} \bar{v}_{jk},$$

where σ_i, u_{ik}, v_{ik} are the non-zero singular values/vectors from large to small, then its pseudo-inverse $A^\dagger$ could be written as,

$$A^\dagger_{ji} = \sigma_k^{-1} u_{ik} \bar{v}_{jk}.$$

Similarly, one could always compute a pseudo-inverse after grouping some of its indices. For $T \in \mathbb{C}^{\mathcal{J}_1 \times \cdots \times \mathcal{J}_k}$, pick rows and columns with $\{s_\ell\}_{1 \le \ell \le R} \sqcup \{t_\ell\}_{1 \le \ell \le C} = \mathcal{J}_1 \times \cdots \times \mathcal{J}_k$ with $R + C = k$, there's a monomorphism (unfold),

$$\phi^{\mathcal{J}_R}_{\mathcal{J}_C} : \mathbb{C}^{\mathcal{J}_1 \times \cdots \times \mathcal{J}_k} \to \mathbb{C}^{\mathcal{J}_R \times \mathcal{J}_C},$$

with $\mathcal{J}_R = \mathcal{J}_{r_1} \times \cdots \times \mathcal{J}_{r_R}$ and $\mathcal{J}_C = \mathcal{J}_{c_1} \times \cdots \times \mathcal{J}_{c_R}$. The pseudo inverse of T induced by $\phi^{\mathcal{J}_R}_{\mathcal{J}_C}$ is just,

$$\left(\phi^{\mathcal{J}_R}_{\mathcal{J}_C}\right)^{-1} \left(\phi^{\mathcal{J}_R}_{\mathcal{J}_C} T\right)^\dagger.$$

For convenience, let's denote $T^\dagger$ with indices $\underline{j}_{r_\ell}$ and $\bar{j}_{c_\ell}$ where underline/overline is just to emphasis it is a row/column index associates with $\phi^{\mathcal{J}_R}_{\mathcal{J}_C}$. Our idea for exercise 1 is to alternatively minimize Frobenius error from T, we claim that this is simply multiplying the pseudo inverse to T.

Claim 1. *Fix $T \in \mathbb{C}^{\mathcal{J}_1 \times \cdots \times \mathcal{J}_k}$, $\{U_{\ell i_\ell}\}_{1 \le i_\ell \le r_\ell}$ for $\ell > 1$, we have,*

$$\arg \min_{\{U_{1 i_1}\}_{1 \le i_1 \le r_1}} \left\| T - \sum_{i_1, \ldots, i_k \in [r_1] \times \cdots \times [r_k]} g_{i_1 \ldots i_k} \bigotimes_{1 \le \ell \le k} U_{\ell i_\ell} \cdot \right\| = T \cdot g^\dagger \prod_{2 \le \ell} U_\ell^\dagger,$$

with

$$\left(T \cdot g^\dagger \prod_{2 \le \ell} U_\ell^\dagger \right)_{i_1 j_1} = T_{j_1 \ldots j_k} \cdot g^\dagger_{\underline{i}_1 \bar{i}_2 \ldots \bar{i}_k} \prod_{2 \le \ell} U^\dagger_{\ell \bar{i}_\ell \underline{j}_\ell}.$$

Notice that the unit-norm condition could be easily satisfied by renormalizing $U_{1 i_1}$ and absorbing multiplicative factors into the core tensor g. Therefore we have the following iteration,

$$\begin{aligned}
\tilde{A}^{(\tau)}_{i_1 j_1} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\underline{i}_1 \bar{i}_2 \bar{i}_3} B^{(\tau)\dagger}_{\underline{i}_2 \bar{j}_2} C^{(\tau)\dagger}_{\underline{i}_3 \bar{j}_3}, & A^{(\tau+1)}_{i_1 j_1} &= \tilde{A}^{(\tau)}_{i_1 j_1} / \|\tilde{A}^{(\tau)}_{i_1}\|, \\
\tilde{B}^{(\tau)}_{i_2 j_2} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\bar{i}_1 \underline{i}_2 \bar{i}_3} C^{(\tau)\dagger}_{\underline{i}_3 \bar{j}_3} A^{(\tau+1)\dagger}_{\underline{i}_1 \bar{j}_1}, & B^{(\tau+1)}_{i_2 j_2} &= \tilde{B}^{(\tau)}_{i_2 j_2} / \|\tilde{B}^{(\tau)}_{i_2}\|, \\
\tilde{C}^{(\tau)}_{i_3 j_3} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\bar{i}_1 \bar{i}_2 \underline{i}_3} A^{(\tau+1)\dagger}_{\underline{i}_1 \bar{j}_1} B^{(\tau+1)\dagger}_{\underline{i}_2 \bar{j}_2}, & C^{(\tau+1)}_{i_3 j_3} &= \tilde{C}^{(\tau)}_{i_3 j_3} / \|\tilde{C}^{\tau}_{i_3}\|, \\
g^{(\tau)}_{i_1 i_2 i_3} &= T_{j_1 j_2 j_3} \cdot A^{(\tau)\dagger}_{\underline{i}_1 \bar{j}_1} B^{(\tau)\dagger}_{\underline{i}_2 \bar{j}_2} C^{(\tau)\dagger}_{\underline{i}_3 \bar{j}_3}.
\end{aligned}$$

For initialization, we pick a seemingly reasonable singular vector with largest singular values. Note that $g^{(0)}$ could be derived from $A^{(0)}, B^{(0)}, C^{(0)}$. Therefore we do not need to initialize g explicitly. However, because initialization for A, B, C are merely heuristic, we also tried to initialize $g^{(0)}$ uniform randomly.

Definition 2. *Given a matrix $A \in \mathbb{C}^{\mathcal{J}_R \times \mathcal{J}_C}$ with SVD,*

$$A_{ij} = \sigma_k u_{ik} \bar{v}_{jk},$$

where σ_i, u_{ik}, v_{ik} are the non-zero singular values/vectors from large to small. Denote its left/right singular vectors as,

$$\mathsf{SVD} A_{\underline{i}j} = u_{ij}\mu_j,$$
$$\mathsf{SVD} A_{i\underline{j}} = v_{ij}\mu_i.$$

with $\mu_k = \begin{cases} 1, & \textit{for } k \le r, \\ 0, & \textit{otherwise.} \end{cases}$

Similarly, we define folded singular vector by,

$$\mathsf{SVD}_{r_1} T := \left(\phi^{\mathcal{J}_R}_{\mathcal{J}_C}\right)^{-1} \mathsf{SVD}_{r_1} \left(\phi^{\mathcal{J}_R}_{\mathcal{J}_C} T\right).$$

For convenience, we might still use underline/overline to emphasis how exactly the indices that is the indices of each singular vectors being collected. Now we could pick,

$$A^{(0)}_{i_1 j_1} = \mathsf{SVD}_{r_1} T_{\underline{i}_1 i_2 i_3},$$
$$B^{(0)}_{i_2 j_2} = \mathsf{SVD}_{r_2} T_{i_1 \underline{i}_2 i_3},$$
$$C^{(0)}_{i_3 j_3} = \mathsf{SVD}_{r_3} T_{i_1 i_2 \underline{i}_3},$$

as initionalization. Next, we consider the Tucker decomposition. Here we use the same initionalization but changes the iteration rules.

Exercise 2. *Set $k = 3$, indices j_ℓ, i_ℓ running in $\mathcal{J}_\ell, [r_\ell]$ respectively. For $T \in \mathbb{C}^{\mathcal{J}}$, find Tucker approximation $A = U_1, B = U_2, C = U_3$,*

$$T_{j_1 j_2 j_3} \approx g_{i_1 i_2 i_3} A_{i_1 j_1} B_{i_2 j_2} C_{i_3 j_3}.$$

First, we could inductively assume that $\{A^{(\tau)}_{i_1}\}_{i_1 \in [r_1]}, \{B^{(\tau)}_{i_2}\}_{i_2 \in [r_2]}, \{C^{(\tau)}_{i_3}\}_{i_3 \in [r_3]}$ are all orthonormal. Therefore any pseudo inverse now becomes componentwise conjugation (with possibly index permutation). In order to keep $A^{(\tau+1)}, B^{(\tau+1)}, C^{(\tau+1)}$ orthonormal, we perform one more SVD to extract those orthonormal componentsm, this would yield the following iteration.

$$\begin{aligned}
\tilde{A}^{(\tau)}_{i_1 j_1} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\underline{i}_1 \bar{i}_2 \bar{i}_3} \bar{B}^{(\tau)}_{i_2 j_2} \bar{C}^{(\tau)}_{i_3 j_3}, & A^{(\tau+1)}_{i_1 j_1} &= \mathsf{SVD}_{r_1} \tilde{A}^{(\tau)}_{\underline{i}_1 j_1}, \\
\tilde{B}^{(\tau)}_{i_2 j_2} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\bar{i}_1 \underline{i}_2 \bar{i}_3} \bar{C}^{(\tau)}_{\underline{i}_3 \bar{j}_3} \bar{A}^{(\tau+1)}_{i_1 j_1}, & B^{(\tau+1)}_{i_2 j_2} &= \mathsf{SVD}_{r_2} B^{(\tau)}_{\underline{i}_2 j_2}, \\
\tilde{C}^{(\tau)}_{i_3 j_3} &= T_{j_1 j_2 j_3} \cdot g^{(\tau)\dagger}_{\bar{i}_1 \bar{i}_2 \underline{i}_3} \bar{A}^{(\tau+1)}_{i_1 j_1} \bar{B}^{(\tau+1)}_{i_2 j_2}, & C^{(\tau+1)}_{i_3 j_3} &= \mathsf{SVD}_{r_3} C^{(\tau)}_{\underline{i}_3 j_3}, \\
g^{(\tau)}_{i_1 i_2 i_3} &= T_{j_1 j_2 j_3} \cdot \bar{A}^{(\tau)}_{i_1 j_1} \bar{B}^{(\tau)}_{i_2 j_2} \bar{C}^{(\tau)}_{i_3 j_3}.
\end{aligned}$$

3 Result and Disscussion

If we restrict $r = r_1 = r_2 = r_3$, the final Frobenius error for both problems are presented in table 1. When increasing the iteration number, as shown in table 1, the error for both exercises decreases when r increases. Furthermore, the although in general the error decreases when iteration number increases 10 iterations has almost reached the optimal. If we instead restrict total number of iteration being 10, the Frobenius error is presented in table 2. In general error decreases when rank increases and giving the extra ranks to different indices would yield slightly different error, which is as expectd. It is worth noting that, although adding in

r	iteration number	Error for 1	Error for 2
1	10	11.97843	11.97843
2	10	10.71081	10.779454
3	10	10.068613	10.150229
4	10	9.444539	9.496887
5	10	9.175108	9.194712
6	10	8.927064	8.983321
7	10	8.605858	8.745392
8	1	8.416768	8.519699
8	10	8.353371	8.50492
8	20	8.350785	8.5030775
8	30	8.350751	8.5030775
8	40	8.350749	8.452305
8	50	8.350748	8.4024
8	60	8.350748	8.42024
8	70	8.350748	8.42024
8	80	8.350748	8.42024
8	90	8.350748	8.42024
8	100	8.350748	8.42024

Table 1: Fixing $r = r_1 = r_2 = r_3$

r_1	r_2	r_3	Error for 1	Error for 2
5	5	5	9.175108	9.194712
8	5	5	8.814871	8.892981
5	8	5	9.019204	9.123034
5	5	8	9.169555	9.1356535
1	5	5	11.113482	11.089706
5	1	5	11.426196	11.344864
5	5	1	11.815943	11.834683

Table 2: Fixing 10 Iteraions

orthonormal constraint, the Tucker decomposition actually performs better than only normalized version. Some extreme cases are presented in table 3. One surprising outcome is that, for $96 \times 5 \times 5$ case in exercise 1, without randomly initialize $g^{(0)}$, the error do not converges to low value. On the other hand, as we randomly initialize $g^{(0)}$, however, error get controlled down to roughly 20. Also, for full sized decomposition, randomly initialize $g^{(0)}$ performs much better than implicitly derive it from the first iteration. We emphasize, however, that this initialization from SVD is only a heuristic and thus not necessarily the optimal setting.

4 Conclusion

In comparison to Tucker decomposition, its relaxed variant and different choices of $g^{(0)}$, we conclude the following. First of all, even though orthogonal constraint in Tucker decomposition might decrease the degree of freedom in its parametrization, it converges better than its non-orthogonal counterpart. Secondly, the error in general decreases as we increases r_1, r_2, r_3. Third, in some extreme cases as we presented previously, it might have difficulty converges to a good minimum. To address this, using orthogonal constraint and initialize core tensor $g^{(0)}$ is recommended.

random initialize	r_1	r_2	r_3	Error for 1	Error for 2
no	96	5	5	8.89	68.11
yes	96	5	5	8.94	19.07
no	96	64	8	1.5e-4	1e-4
yes	96	64	8	2.4e-13	6.35e-8

Table 3: Others (Iterate Until Extrema)