Tensor Algebra
University Federal do Ceará
PPGETI - 2020.2
Saulo Mendes de Melo

Homework 7

High Order Singular Value Decomposition (HOSVD)

1 Problem 1

For a third-order tensor $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$ implement the truncated high-order singular value decomposition (HOSVD), using the following prototype function:

$$[\mathcal{S}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}] = hosvd(\mathcal{X}) \tag{1}$$

Solution

This work uses functions of the *Tensorly* package for *Python* for tensor operations, like the n-mode product and unfolding. In Listing 1 we declare a tensor $\mathcal{X} \in \mathbb{C}^{3\times4\times2}$ for testing purposes. The shape definition looks wrong, but it is done like that due to the way *numpy* defines 3-dimensional tensors. They are defined as arrays of matrices, so the tube fibers dimension comes first.

```
np.random.seed(0)
shape = (2, 3, 4)

X = randn(shape[0], shape[1], shape[2])

np.random.seed(1)
X = X + randn(shape[0], shape[1], shape[2])*1j
```

Listing 1: Creating $\mathcal{X} \in \mathbb{C}^{3\times4\times2}$

The Listing 2 contains the definition of our HOSVD and a custom Hermitian function. This function is merely a convenience for cleaner code. The HOSVD function takes the approach of calculating the SVD of all of the matrix unfoldings of input tensor $\mathcal{X}$, so for each n-mode unfolding of $\mathcal{X}$, we take the left singular vectors of the unfolded matrix and save it as $\mathbf{U}^{(n)}$. $\mathcal{X}$ is then calculated using equation 2.

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}^{(1)H} \times_2 \mathbf{U}^{(2)H} \times_3 \cdots \times_N \mathbf{U}^{(N)H} \tag{2}$$

The HOSVD function also has a input parameter for a custom multilinear-rank, as to calculate the Truncated form of the HOSVD. The assertion at the end of the block guarantees the correctness of the method by assuring that equation 3 is true.

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \cdots \times_N \mathbf{U}^{(N)} \tag{3}$$

```
def Hermitian(X):
    return np.array(np.matrix(X).H)

def HOSVD(X, ml_rank=None):
    U_list = []
    S = X

    for n in range(X.ndim):
        X_n = tl.unfold(X, n)
        Un, _, _ = svd(X_n)

        if ml_rank:
            Un = Un[:, :ml_rank[n]]

        U_list.append(Un)

        S = mode_dot(S, Hermitian(Un), n)

    return S, U_list


S, U = HOSVD(X)
assert X.all() == mode_dot(mode_dot(mode_dot(S, U[0], 0), U[1], 1), U[2], 2).all()
```

Listing 2: Definition of the HOSVD method

2 Problem 2

Consider the two third-order tensors $\mathcal{X} \in \mathbb{C}^{8\times4\times10}$ and $\mathcal{Y} \in \mathbb{C}^{5\times5\times5}$ provided in the file "hosvd_denoising.mat". By using your HOSVD prototype function, find a low multilinear rank approximation for these tensors, defined as $\tilde{\mathcal{X}} \in \mathbb{C}^{R_1\times R_2\times R_3}$ and $\tilde{\mathcal{Y}} \in \mathbb{C}^{P_1\times P_2\times P_3}$. Then, calculate the normalized mean square error (NMSE) between the original tensor and its approximation, i.e.,:

$$\text{NMSE}(\tilde{\mathcal{X}}) = \frac{\|\tilde{\mathcal{X}} - \mathcal{X}\|_F^2}{\|\tilde{\mathcal{X}}\|_F^2}, \quad \text{NMSE}(\tilde{\mathcal{Y}}) = \frac{\|\tilde{\mathcal{Y}} - \mathcal{Y}\|_F^2}{\|\tilde{\mathcal{Y}}\|_F^2}$$

Hint: The multilinear ranks of $\mathcal{X}$ and $\mathcal{Y}$ can be found by analysing the profile of the 1-mode, 2-mode and 3-mode singular values of these tensors.

Solution

To find the best low multilinear rank approximation for tensors $\mathcal{X}$ and $\mathcal{Y}$, an analysis of the singular values was conducted. The HOSVD was computed for $\mathcal{X}$ and $\mathcal{Y}$, resulting in $\mathcal{S}_{\mathcal{X}}$ and $\mathcal{S}_{\mathcal{Y}}$, which had their respective $\sigma_i^{(n)} = \|\mathcal{S}_{i_n=i}\|_F$ calculated. Figure 1 plots the singular values for HOSVD($\mathcal{X}$). There is a clear pattern of the first larger singular values progressively descending down to a point of relative stability. Judging by this graph, good choice for a low multilinear rank for HOSVD would be $(R_1 = 5, R_2 = 2, R_3 = 3)$.

Figure 1: Singular values vs. index for $\mathcal{X}$

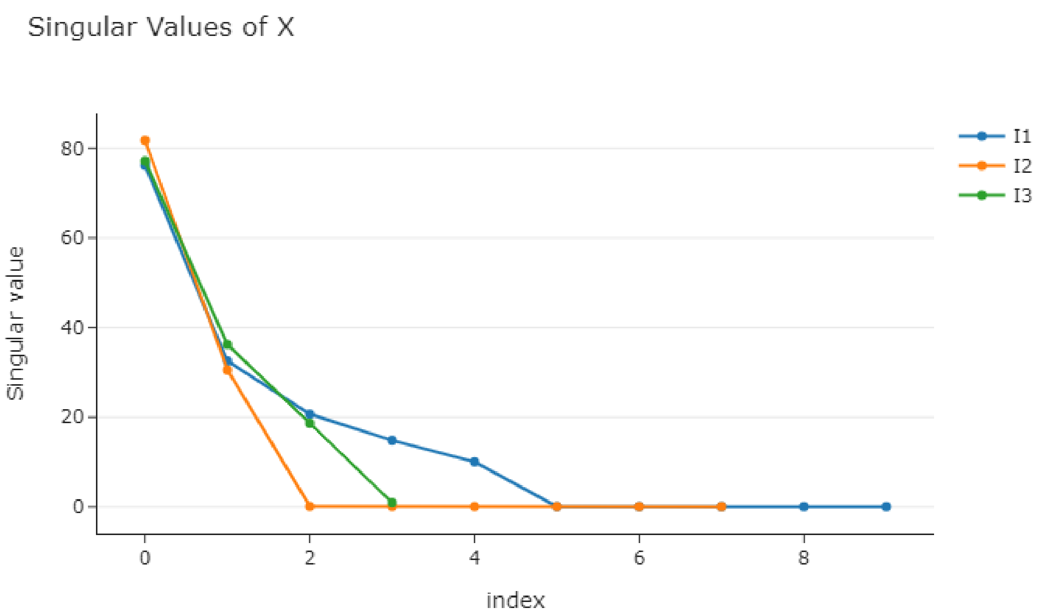
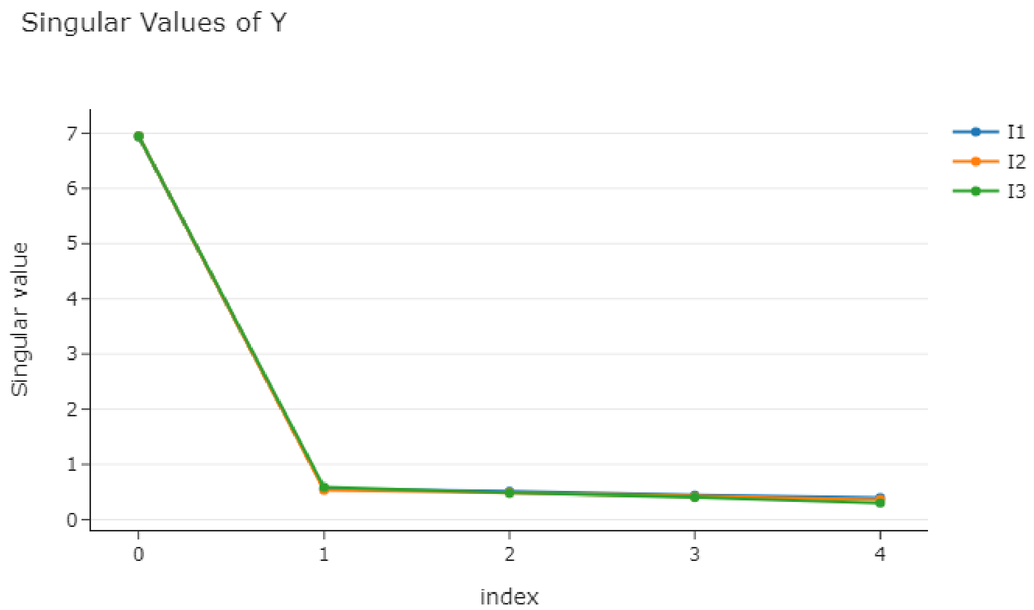


Figure 2 show the singular value behavior for $\mathcal{Y}$. There is a similar behavior to Figure 1, but in this case all singular values converge in $(P_1 = 1, P_1 = 1, P_1 = 1)$.

Figure 2: Singular values vs. index for $\mathcal{Y}$



Computing the truncated HOSVD for tensors $\mathcal{X}$ and $\mathcal{Y}$ with the respective low multilinear ranks obtained, we can construct estimated tensors $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$. The resulting errors are $\text{NMSE}(\tilde{\mathcal{X}}) = 0.00012$ and $\text{NMSE}(\tilde{\mathcal{Y}}) = 0.02047$, which are relatively low.

3 Extra Problem

Implement a prototype function for the Higher-Order Orthogonal Iteration (HOOI).

Solution

Listing 3 contains the implementation of the HOOI. It starts by taking an initialization of $\mathbf{U}$ from the HOSVD of the input tensor. Then, for each n-mode it computes $\tilde{\mathcal{U}}_n = \mathcal{X} \times_1 \mathbf{U}^{(1)} \cdots \times_{n-1} \mathbf{U}^{(n-1)} \times_{n+1} \mathbf{U}^{(n+1)} \cdots \times_N \mathbf{U}^{(N)}$. After that, $\mathbf{U}^{(n)}$ is set as the leading left singular vectors of n-mode unfolding $[\tilde{\mathcal{U}}_n]_{(n)}$. The singular values $\mathcal{S}$ are computed via (2).

```python
def HOOI(X):
    _, U = HOSVD(X)

    U_list = []
    for i in range(X.ndim):

        Un_tilde = X
        for j, Un in enumerate(U):
            if j != i:
                Un_tilde = mode_dot(Un_tilde, Hermitian(Un), j)
            else:
                continue

        Un_f, _, _ = svd(tl.unfold(Un_tilde, i))

        U_list.append(Un_f)

    S = X
    for k, Un in enumerate(U_list):
        S = mode_dot(S, Hermitian(Un), k)

    return S, U_list
```

Listing 3: Definition of the HOOI method