University of Vienna
Faculty of Mathematics

TENSOR METHODS FOR DATA SCIENCE AND SCIENTIFIC COMPUTING

Milutin Popovic

December 8, 2021

Contents

1 Assignment 4

1.1 HOSVD Algorithm

The HOSVD algorithm is used to compute the Tucker approximation of a given tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with given ranks $r_1, \dots, r_d \ \in \mathbb{N}$ ($d \in \mathbb{N}$ and $n_1, \dots, n_d \ \in \mathbb{N}$. Additionally we would like to compute and save

- the Forbenius norms of the error produced at each step of the algorithm $\frac{\|A-\hat{A}_k\|_F}{\|A\|_F}$ and the vectors of
- the vector of singular values directly approximated by the algorithm

The Tucker decomposition for A, with the ranks above is the following

$$A_{i_1,\cdots,i_d} = \sum_{\alpha_1=1}^{r_1} \cdot \sum_{\alpha_d=1}^{r_d} (U_1)_{i_1,\alpha_1} \cdots (U_d)_{i_d,\alpha_d} S_{\alpha_1,\dots,\alpha_d}, \tag{1}$$

where $U_k \in \mathbb{R}^{n_k \times r_k}$ for all $k \in \{1, \cdots, d\}$ and $S \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is called the Tucker-core.

The HOSVD algorithm with the additional requirements for the Tucker decomposition of A is the following

Algorithm 1 HOSVD algorithm

$\hat{S}_0 \leftarrow A$
$A_0 \leftarrow A$
for $k = 1, \dots, d$ **do**
  $(B_k)_{\alpha_1 \cdots \alpha_{k-1} \cdot i_{k+1} \cdots i_d, i_k} \leftarrow (\hat{S}_{k-1})_{\alpha_1, \dots, \alpha_{k-1}, i_k, \dots, i_d}$ ▷ permute then reshape
  $\hat{B}_k \leftarrow \hat{U}_k \hat{\Sigma}_k \hat{V}_k^*$ ▷ rank r_k T-SVD for B_k
  $(\hat{S}_k)_{\alpha_1, \dots, \alpha_k, i_{k+1}, \dots, i_d} \leftarrow (B_k \hat{V}_k)_{\alpha_1, \dots, \alpha_{k-1}, i_{k+1}, \dots, i_d, \alpha_k}$ ▷ reshape then permute
  $(A_k)_{i_1, \cdots, i_d} \leftarrow \sum_{\alpha_1=1}^{r_1} \cdot \sum_{\alpha_k=1}^{r_k} (\hat{V}_1)_{i_1, \alpha_1} \cdots (\hat{V}_k)_{i_d, \alpha_d} S_{\alpha_1, \dots, \alpha_k, i_{k+1}, \dots, i_d}$
  save $\frac{\|A - \hat{A}_k\|_F}{\|A\|_F}$
  save $\hat{V}_k$
  save $\hat{\Sigma}_k$
end for

We note that at the $k-th$ step of the algorithm the shapes of the tensors are

$$\hat{S}_{k-1} \in \mathbb{R}^{r_1 \times \cdots \times r_{k-1} \times n_k \times \cdots n_d} \tag{2}$$

$$\hat{V}_k \in \mathbb{R}^{n_k \times r_k}, \tag{3}$$

$$\hat{B}_k \in \mathbb{R}^{r_1 \cdots r_{k-1} \cdot n_{k+1} \cdots n_d, \times n_k}, \tag{4}$$

$$\hat{\Sigma}_k \in \mathbb{R}^{r_k \times 1}. \tag{5}$$

1.2 Testing the HOSVD

For the case $d = 4$ we construct a quasirandom Tucker decomposition by drawing the entries for $U_k \in \mathbb{R}^{n_k \times r_k}$ and $S \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ uniformly on $[-1, 1]$. The output of the errors in the $k-th$ steps is in the figure bellow

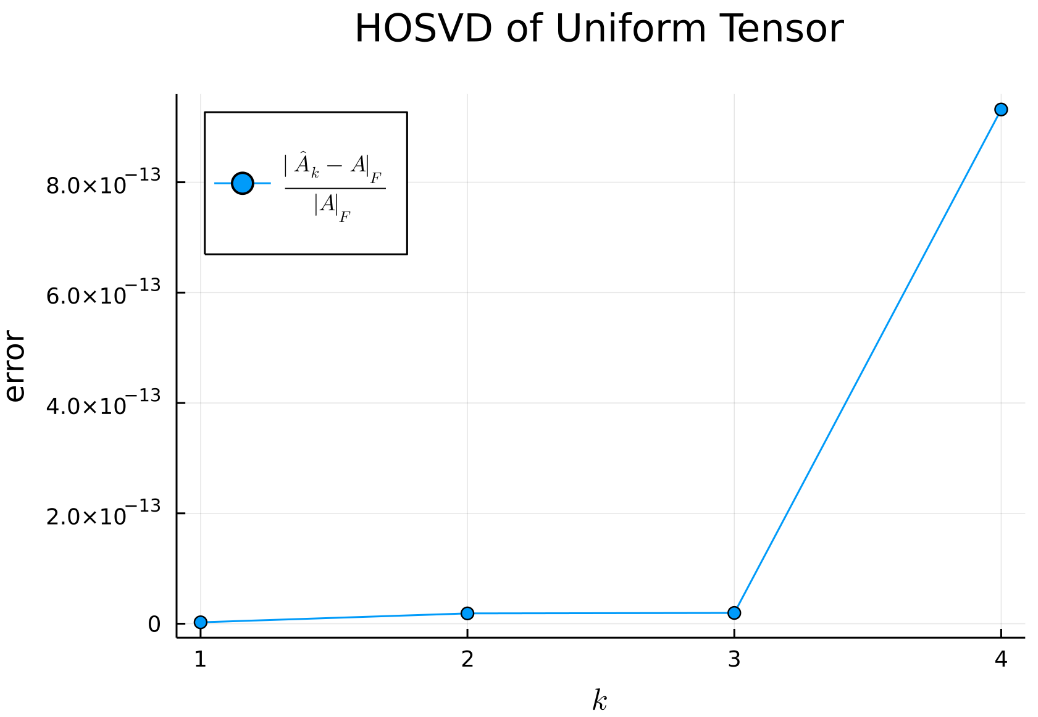

Figure 1: Tucker approximation error on the $k-th$ step of a quasirandom Tensor

1.3 Tucker approximation of function-related tensors

Consider two multivariable functions $f(x_1,\dots,x_d)$ and $g(x_1,\dots,x_d)$, defined as

$$f(x_1,\dots,x_d) = \left(1+\sum_{k=1}^{d}\frac{x_k^2}{8^{k-1}}\right)^{-1} \tag{6}$$

$$g(x_1,\dots,x_d) = \sqrt{\sum_{k=1}^{d}\frac{x_k^2}{8^{k-1}}}\cdot\left(1+\frac{1}{2}\cos\left(\sum_{k=1}^{d}\frac{4\pi x_k}{4^{k-1}}\right)\right) \tag{7}$$

for $x_1,\dots,x_d \in [-1,1]$. Additionally we define a grid of points

$$t_i = 2\frac{i-1}{n-1}-1, \tag{8}$$

for $i=1,\dots,n$. With this we can construct a d dimensional tensor of size $n\times\cdots\times n$ by

$$b_{i_1,\cdots,i_d} = f(t_{i_1},\dots,t_{i_d}), \tag{9}$$
$$a_{i_1,\cdots,i_d} = g(t_{i_1},\dots,t_{i_d}), \tag{10}$$

for all $i_1,\cdots,i_d \in \{1,\cdots,n\}$.

For $C=A$ and $C=B$. For every $k\in\{1,\dots,d\}$, we compute the singular values of the $k-th$ Tucker unfolding matrix of C.

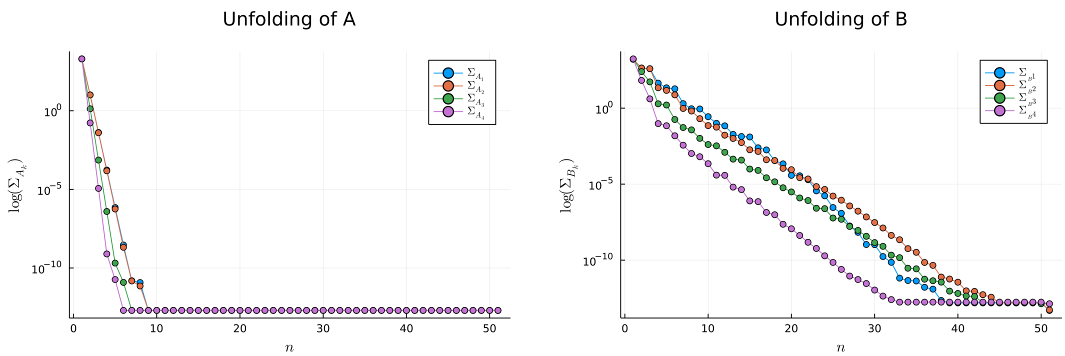

Figure 2: Decrease of singular values of the $k-th$ Tucker unfolding of A, B produced by its SVD

For the accuracy Threshold $\varepsilon_j = 10^{-j}$ with $j\in\{2,4,\dots,12\}$, for each j and every k we find the smallest r_{jk} such that the $k-th$ unfolding matrix can be approximated with ranks r_{jk}, where the approximations relative error does not exceed ε_j. Additionally we compute the ε_{jk} error in the $k-th$ unfolding, which should by the error analysis be bounded by $\varepsilon_{jk}\le\varepsilon_j$ for all j.

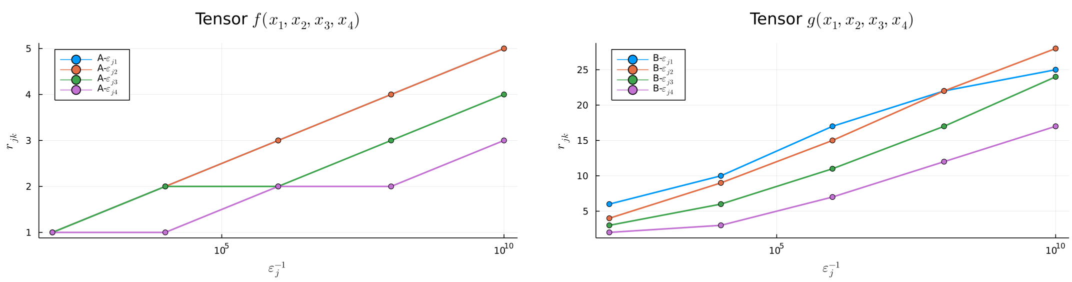

Figure 3: Dependence of the approximated ranks r_{jk} on $j=\log_{10}\varepsilon^{-j}$

For every j we use our implementation of the HOSVD algorithm to compute the Tucker approximation of C for the ranks r_{jk} for $k=1,\cdots,d$ and the number of total entries N_j produced by the output decomposition. In the code [1] it is checked during run-time that the error produced by the HOSVD algorithm does not exceed $\|\varepsilon_{jk}\|_F$ and agrees wit the error analysis.

For $j = 12$ we plot the ratio of the singular values produced by the HOSVD algorithm and during the SVD approximation of the $k-th$ Tucker unfolding matrix

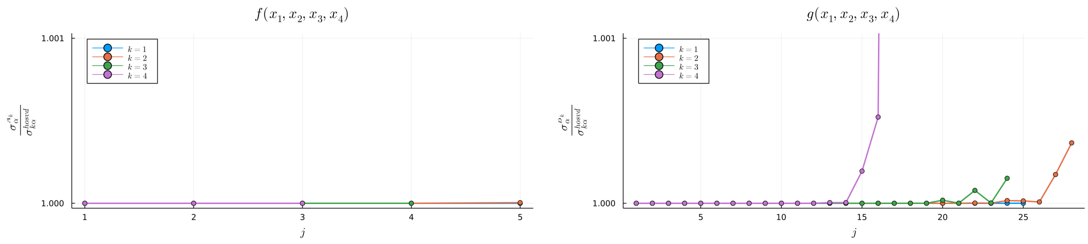

Figure 4: Ratio of Singular values produced by the HOSVD and the $k-th$ Tucker unfolding approximation of C.

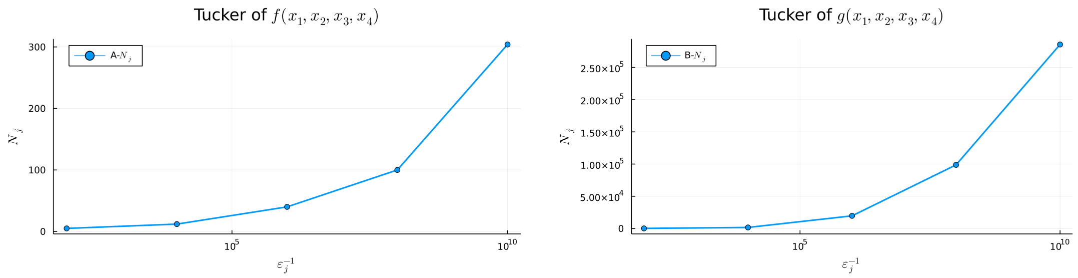

Figure 5: Number of parameters N_j produced by the HOSVD for C

1.4 TT-SVD for MPS-TT

The TT-MPS (Tensor Train or Matrix Product States decomposition) for a given tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with ranks $r_1, \ldots, r_{d-1} \in \mathbb{N}$ is the following

$$A_{i_1,\ldots,i_d} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} U_1(\alpha_0, i_1, \alpha_1) \cdots U_d(\alpha_{d-1}, i_d, \alpha_d) \tag{11}$$

for $\alpha_0 = \alpha_d = 1$, $i_k \in \{1, \ldots, n_k\}$. The decomposition factors are given by

$$V_k(i_k) \in \mathbb{R}^{r_{k-1} \cdot n_k \times r_k}, \tag{12}$$
$$(V_k(i_k))_{\alpha_{k-1},\alpha_k} = U_k(\alpha_{k-1}, i_k, \alpha_k), \tag{13}$$

where i_k is called the mode index.

The TT-SVD algorithm for A as above is the following

Algorithm 2 TT-SVD algorithm

$\hat{S}_0 \leftarrow A$
$A_0 \leftarrow A$
$r_0 \leftarrow 1$
$r_d \leftarrow 1$
for $k = 1, \ldots, d-1$ **do**
  $(B_k)_{\alpha_{k-1} \cdot i_k, i_{k+1} \cdots i_d} \leftarrow (\hat{S}_{k-1})_{\alpha_{k-1}, i_k, i_{k+1}, \ldots, i_d}$ ▷ reshape
  $\hat{B}_k \leftarrow \hat{U}_k \hat{\Sigma}_k \hat{V}_k^*$ ▷ rank r_k T-SVD for B_k
  $(\hat{C}_k)_{\alpha_{k-1}, i_k, \alpha_k} \leftarrow (\hat{U}_k)_{\alpha_{k-1} i_k, \alpha_k}$ ▷ reshape
  $(\hat{S}_k)_{\alpha_k, i_k, \ldots, i_d} \leftarrow (\hat{\Sigma}_k \hat{V}_k)_{\alpha_k, i_{k+1} \ldots i_d}$ ▷ reshape
  save $\frac{\|A - \hat{A}_k\|_F}{\|A\|_F}$
  save $\hat{C}_k$
  save $\hat{\Sigma}_k$
end for

1.5 Testing the TT-SVD

For the case $d = 4$ we construct a quasirandom Tucker decomposition by drawing the entries for $C_k \in \mathbb{R}^{r_{k-1} \cdot n_k \times r_k}$ uniformly on $[-1,1]$. The output of the errors in the $k-th$ steps is in the figure bellow

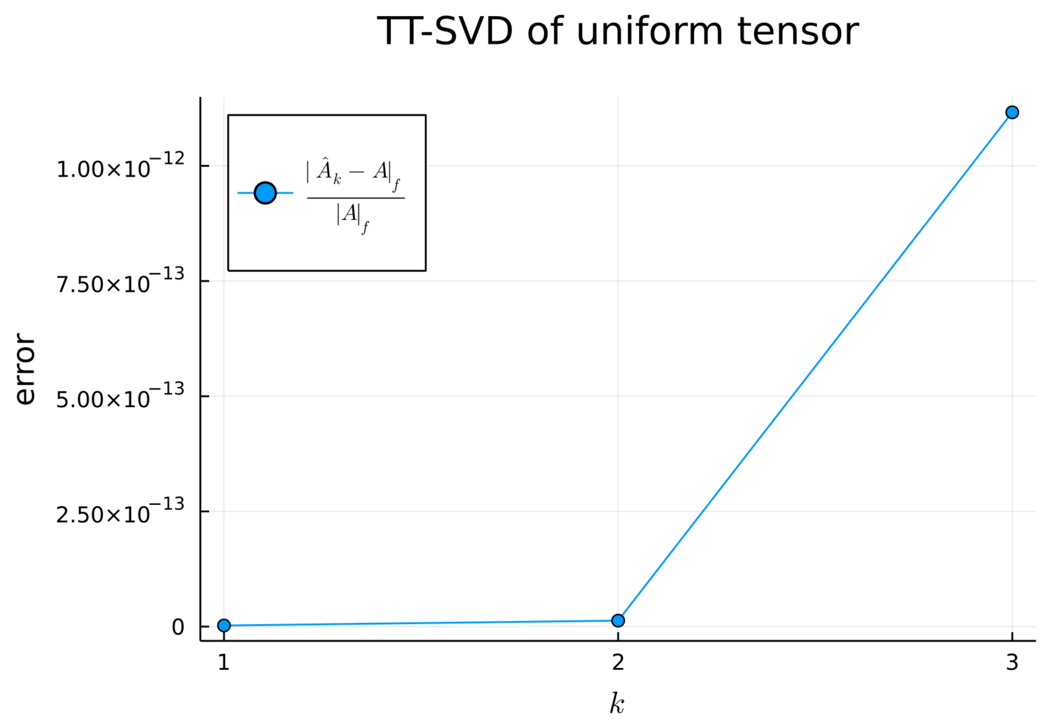

Figure 6: TT-MPS error on the $k-th$ step of a quasirandom Tensor in the TT-SVD algorithm

1.6 TT-MPS of function-related tensors

In this section we repeat everything we did in section 1.3, replacing the HOSVD algorithm for the Tucker approximation with the TT-SVD algorithm for the TT-MPS approximation

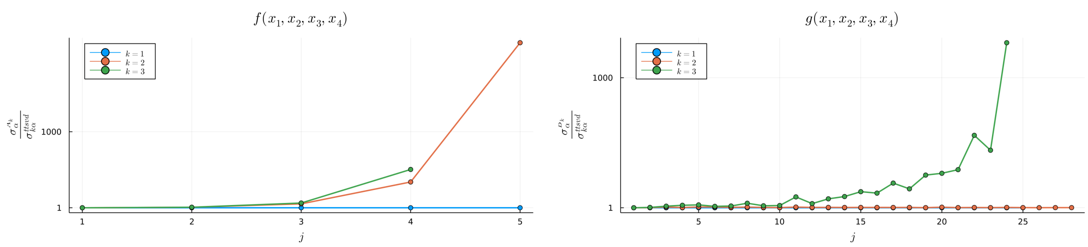

Figure 7: Ratio of Singular values produced by the TT-SVD and the $k-th$ Tucker unfolding approximation of C.

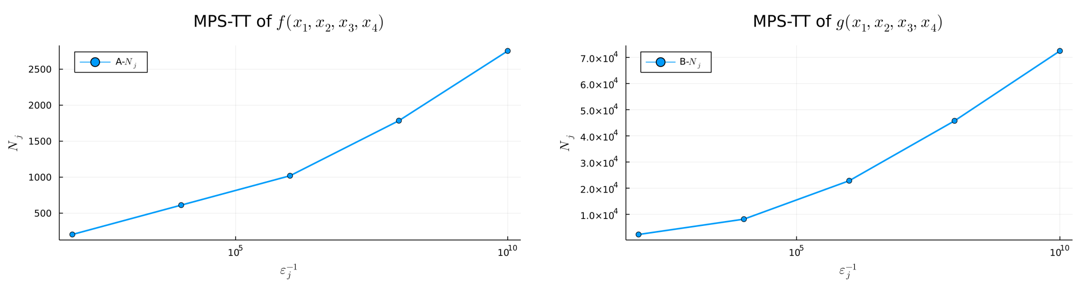

Figure 8: Number of parameters N_j produced by the TT-SVD for C

References

[1] Popovic Milutin. *Git Instance, Tensor Methods for Data Science and Scientific Computing*. URL: git://popovic.xyz/tensor_methods.git.