250085 VU
TENSOR METHODS FOR DATA SCIENCE AND SCIENTIFIC COMPUTING
WINTER SEMESTER 2021

HOMEWORK ASSIGNMENT 4

VLADIMIR KAZEEV

Due by **13:00 on Monday, 6th December 2021**. To be submitted via Moodle.

The theoretical part may be (i) typeset, (ii) handwritten digitally or (iii) handwritten on a physical medium and then digitized by scanning or photographing. For the programming part, any programming language and environment may be used.

For each student submitting a solution, the submission should consist of (i) a single PDF file presenting the solution of the theoretical part and the results of the programming part in a self-contained fashion (so that running the student's code not be required for understanding the results) and (ii) the code (if any), ready to run and reproduce the results presented in the PDF file, organized in any reasonable number of files.

In this assignment, we consider the following functions f and g defined pointwise on $[-1,1]^d$ with $d \in \mathbb{N}$ as follows:

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

for all $x_1,\dots,x_d \in [-1,1]$.

For $n \in \mathbb{N}$, we consider a grid of the points

$$t_i = 2\,\frac{i-1}{n-1} - 1 \quad \text{with} \quad i = 1,\dots,n$$

and the d-dimensional tensors A and B of size $n \times \cdots \times n$ composed of the values of f and g at the corresponding grid points:

$$a_{i_1,\dots,i_d} = f(t_{i_1},\dots,t_{i_d}) \quad \text{and} \quad b_{i_1,\dots,i_d} = g(t_{i_1},\dots,t_{i_d}) \quad \text{for all} \quad i_1,\dots,i_d \in \{1,\dots,n\}\,.$$

1. Implementing the HOSVD algorithm.

Implement the HOSVD algorithm for the Tucker approximation of a given tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ (with dimensionality $d \in \mathbb{N}$ and mode sizes $n_1,\dots,n_d \in \mathbb{N}$) with given ranks $r_1,\dots,r_d \in \mathbb{N}$. The output should include (i) the Tucker decomposition produced by the algorithm, (ii) the vectors of singular values of the matrices that are explicitly approximated within the algorithm (one vector of singular values per step) and (iii) the Frobenius norms of the errors of the mentioned low-rank matrix approximation (one scalar per step).

2. Testing the HOSVD algorithm.

Test your implementation of the HOSVD algorithm in the case of $d = 4$ as follows. For $n_k = 20 + k$ and $r_k = 2k$ with $k \in \{1,\dots,d\}$, generate a quasi-random Tucker decomposition $U_1,\dots,U_d,S$ of mode sizes $n_1,\dots,n_d$ and ranks $r_1,\dots,r_d$ by drawing the entries of the factors

quasi-randomly in $[-1,1]$ ("from the uniform distribution on $[-1,1]$"). Form the tensor C represented by the decomposition. Then compute a Tucker decomposition of C with ranks $r_1,\dots,r_d$ using your implementation of the HOSVD algorithm. Form the tensor $\widehat{C}$ represented by the output decomposition and compute the relative error $\|\widehat{C}-C\|_{\mathrm{F}}/\|C\|_{\mathrm{F}}$.

Hint: the function from Assignment 2 implementing mode-k contraction may be used to form a tensor from its Tucker decomposition.

3. Tucker approximation for function-related tensors.

Consider the tensors A and B defined above in the case of $d=4$ and $n=51$. Do the following for $C=A$ and for $C=B$.

a) For every $k\in\{1,\dots,d\}$, compute the singular values of the kth Tucker unfolding matrix of C. Show the decay of the singular values of all unfolding matrices in a single plot.

Consider the accuracy thresholds $\varepsilon_j=10^{-j}$ with $j\in\{2,4,\dots,12\}$. For each k and every j, find the smallest r_{jk} such that the kth unfolding matrix can be approximated with rank r_{kj} and with *relative* Frobenius-norm error not exceeding ε_j and compute the respective *relative* approximation error ε_{jk} ($\varepsilon_{jk}\le\varepsilon_j$ should therefore hold for all k and j).

In a single plot for all $k\in\{1,\dots,d\}$, show the dependence of r_{jk} on $j=\log_{10}\varepsilon_j^{-1}$.

Referring to your observations and to the definition of f and g, qualitatively explain the dependence of r_{jk} on j for every fixed k and on k for every fixed j.

b) For each j, use your implementation of the HOSVD algorithm to compute an approximate Tucker decomposition of C with ranks $r_1,\dots,r_d$ and compute the number N_j of the parameters of the output decomposition (the total number of entries of all factors). Form the respective approximation $\widehat{C}$ to C by evaluating the decomposition and compute the relative error $\widehat{\varepsilon}_j=\|\widehat{C}-C\|_{\mathrm{F}}/\|C\|_{\mathrm{F}}$.

Check that this error does not exceed $\sqrt{\sum_{k=1}^{d}\varepsilon_{jk}^2}$ and therefore agrees with the error analysis developed in the lectures.

c) For $j=12$ and for every k, plot the dependence, on index α, of **the ratios of** the αth singular value $\sigma_{k,\alpha}^{\mathrm{HOSVD}}$ of the kth unfolding matrix of C **to** the αth singular value $\sigma_{k,\alpha}$ of the matrix that the HOSVD algorithm approximates at step k (which were required above to be returned by the implementation). What relation do you observe? What part of the error analysis given in the lectures can be used to explain your observation and (in a couple of lines) how?

d) Plot the dependence of N_j vs. $j=\log_{10}\varepsilon_j^{-1}$ in log-log scale. What asymptotic behavior do you observe and what convergence of the error with respect to the number of parameters does it suggest in general?

4. Evaluating a MPS-TT representation.

Implement a function that, for $d\in\mathbb{N}$ and $n_1,\dots,n_d\in\mathbb{N}$, evaluates a given MPS-TT decomposition with $d\in\mathbb{N}$ factors $U_1,\dots,U_d$ and returns the tensor A represented by the decomposition.

5. Implementing the Schmidt decomposition (TT-SVD) algorithm.

Implement the TT-SVD (the Schmidt decomposition) algorithm for the MPS-TT approximation of a given tensor $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ (where $d \in \mathbb{N}$ and $n_1, \ldots, n_d \in \mathbb{N}$) with given ranks $r_1, \ldots, r_{d-1} \in \mathbb{N}$. The output should include (i) the MPS-TT decomposition produced by the algorithm, (ii) the vectors of singular values of the matrices that are explicitly approximated within the algorithm (one vector of singular values per step) and (iii) the Frobenius norms of the errors of the mentioned low-rank matrix approximation (one scalar per step).

6. Testing the Schmidt decomposition (TT-SVD) algorithm.

Test your implementation of the Schmidt decomposition (TT-SVD algorithm) in the case of $d = 4$ as follows. For $n_k = 20 + k$ with $k \in \{1, \ldots, d\}$ and $r_k = 2k$ with $k \in \{1, \ldots, d-1\}$, generate a quasi-random MPS-TT decomposition $U_1, \ldots, U_d$ of mode sizes $n_1, \ldots, n_d$ and ranks $r_1, \ldots, r_{d-1}$ by drawing the entries of the factors quasi-randomly in $[-1, 1]$ ("from the uniform distribution on $[-1, 1]$"). Form the tensor C represented by the decomposition. Then compute an MPS-TT decomposition of C with ranks $r_1, \ldots, r_{d-1}$ using your implementation of the Schmidt decomposition (TT-SVD) algorithm. Form the tensor $\widehat{C}$ represented by the output decomposition and compute the relative error $\|\widehat{C} - C\|_{\mathrm{F}} / \|C\|_{\mathrm{F}}$.

7. MPS-TT approximation for function-related tensors. Repeat the steps of Problem **3.** for the MPS-TT decomposition and the TT-SVD algorithm in place of the Tucker decomposition and the HOSVD algorithm. Note that index k, enumerating the steps of the TT-SVD algorithm in this case, varies from 1 to $d - 1$ only.