University of Vienna
Faculty of Mathematics

TENSOR METHODS FOR DATA SCIENCE AND SCIENTIFIC COMPUTING

Milutin Popovic

November 11, 2021

Contents

1 Assignment 3

1.1 Implementing the CP-ALS Algorithm

The main idea of the algorithm is that we have a rank $R \in \mathbb{N}$ CPD of a tensor $A \in \mathbb{R}^{n_1 \times \cdot n_d}$ given in terms of $U_k \in \mathbb{R}^{n_k \times R}$, for $\mathbb{N} \ni k > 2$, $k = \{1,\dots,d\}$ and for $n_1,\dots,n_d \in \mathbb{N}$. We want to find a rank $r \in \mathbb{N}$ CPD of the tensor A, by taking an initial guess for some $V_k \in \mathbb{R}^{n_k \times r}$ and updating it for each k by optimizing the problem

$$\phi(V_1,\dots,V_d) \to \min_{V_k \in \mathbb{R}^{n_k \times r}}, \tag{1}$$

where $\phi(V_1,\dots,V_d)$ is the error function, determined by

$$\phi(V_1,\dots,V_d) = \|\Psi_r(V_1,\dots,V_d) - \Psi_R(U_1,\dots,U_d)\|_F\,. \tag{2}$$

For Ψ_r and Ψ_R denote the CPD multilinear representation maps transforming the CP decomposition into the tensor

$$U = \Psi_R(U_1,\dots,U_d) \tag{3}$$
$$V = \Psi_r(V_1,\dots,V_d) \tag{4}$$
$$\tag{5}$$

for $k = 1,\dots d, d-1,\dots 2$ sequentially by updating V_k at each step of the optimization. **This is one iteration step**.

For Ψ we will use the implementation constructed in the last exercise, which consists of applying the Kronecker product of the rows and then summing over them. Once we define the following matrices we may rewrite the optimality condition in a linear system which is solvable since it is a least square problem.

We define matrices $\mathscr{V}_k \in \mathbb{R}^{n_1 \cdots n_d \times n_k \cdot r}$ and $\mathscr{U}_k \in \mathbb{R}^{n_1 \cdots n_d \times n_k \cdot R}$ for $k \in \{1, \dots, d\}$ by the following

$$\mathscr{U}_k = \prod_{\substack{l=1 \\ l!=k}}^{d} U_l \otimes \mathbb{1}_{\mathbb{R}^{n_k \times n_k}}, \tag{6}$$

$$\mathscr{V}_k = \prod_{\substack{l=1 \\ l!=k}}^{d} V_l \otimes \mathbb{1}_{\mathbb{R}^{n_k \times n_k}}, \tag{7}$$

note that $\prod$ represents the elementwise matrix multiplication and $\otimes$ the Kronecker product. The product of these two new defined matrices is then simply

$$\mathscr{V}_k^* \mathscr{V}_k = \prod_{\substack{l=1 \\ l!=k}}^{d} (V_l^* V_l) \otimes \mathbb{1}_{\mathbb{R}^{n_k \times n_k}} \tag{8}$$

$$\mathscr{V}_k^* \mathscr{U}_k = \prod_{\substack{l=1 \\ l!=k}}^{d} (V_l^* U_l) \otimes \mathbb{1}_{\mathbb{R}^{n_k \times n_k}}. \tag{9}$$

Do note that $\mathscr{V}_k^* \mathscr{V}_k \in \mathbb{R}^{r \cdot n_k \times r \cdot n_k}$ and $\mathscr{V}_k^* \mathscr{U}_k \in \mathbb{R}^{r \cdot n_k \times R \cdot n_k}$. For convenience let us define $F_k := V_k^* U_k$ and $G_k := V_k^* V_k$, which will become useful when talking about storing the computation. The we can solve for V_k with the following least square problem

$$(\mathscr{V}_k^* \mathscr{V}_k) \hat{v}_k = (\mathscr{V}_k^* \mathscr{U}_k) u_k, \tag{10}$$

where $\hat{v}_k = \text{vec}(\hat{V}_k)$ and $u_k = \text{vec}(U_k)$.

To make the computation cost linear with respect to d, we can compute $\mathscr{V}_k^* \mathscr{V}_k$ and $\mathscr{V}_k^* \mathscr{U}_k$ for each k and update them in the iteration in k with the new $\hat{V}_k$, by computing the product in equations 8 and 9 again. Additionally we compute the error ϕ and $\|\nabla \frac{1}{2} \phi^2\|_2$ after each iteration (after going through $k = 2, \dots, d, d-1, \dots, 1$). The implementation is in [1].

1.2 CP-ALS for the matrix multiplication tensor

In this section we will use the matrix multiplication tensor to play around the CP-ALS algorithm we implemented in the last section. We consider $n = 2$ and $r = 7$, $n = 3$ and $r = 23$, $n = 4$ and $r = 49$ for the multiplication tensor and its CP decomposition. The implementation of them we have done in the last exercise together with their rank $R = n^3$ CP decomposition. We test the our implementation of the CP-ALS algorithm of these three multiplication tensor for three random initial guesses V_1, V_2, V_3, where each matrix has unitary columns (i.e. of norm one).

Our procedure will be that we do seven different guesses if they are good enough, i.e. if they do not produce a *SingularValueError* after 10 iterations we go for 10000 iterations and plot both ϕ and $\|\nabla \frac{1}{2} \phi\|_2$ with respect to the number of iterations for each multiplication tensor. We get the following curves

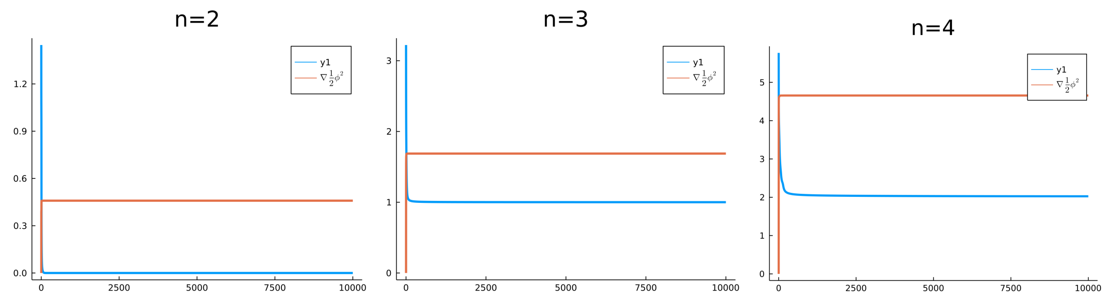

Figure 1: From left to right, rank $r \in \{7, 23, 49\}$ CP-ALS error for the $n \in \{2, 3, 4\}$ multiplication tensor at 10000 iteration steps

References

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