Project Report
MATH-403 LRAT
Code Repository

Yehao Liu Mattia Mariantoni

December 2020

Abstract

The goal of this project is to develop a modification of the HOSVD algorithm which is based on the randomized algorithm. The modified HOSVD should be of the following form:

1. **procedure** mHOSVD($\mathcal{X}$)
2.   **for** $\mu = 1, 2, 3$ do
3. $\tilde{U}_\mu \leftarrow$ orthonormal basis for approximation of range of $X^{(\mu)}$
4.   **end for**
5. $\tilde{\mathcal{C}} \leftarrow \tilde{U}_1^T \circ_1 \tilde{U}_2^T \circ_2 \tilde{U}_3^T \circ_3 \mathcal{X}$
6. $[\mathcal{C}, V_1, V_2, V_3] = \text{HOSVD}(\tilde{\mathcal{C}})$
7.   **for** $\mu = 1, 2, 3$ **do**
8. $U_\mu \leftarrow \tilde{U}_\mu V_\mu$
9.   **end for**
10. **end procedure**

Lines $6 - 9$ are only necessary if the basis matrices from Line 3 and the core tensor from Line 5 are unnecessary large, for example when there is an oversampling parameter involved in their calculation. If their size is optimal, Lines $6 - 9$ can be skipped.

1 Algorithm

1.1 Task 1

We implemented the Randomized Range Finder in RRF.m and the Adaptive Randomized Range Finder ARRF.m following the pseudo-code of [1].
Regarding RRF we take as input an $m \times n$ matrix A and an integer L, this algorithm computes an $m \times L$ orthonormal matrix Q whose range approximates the range of A with rank L.
About ARRF we take as input an $m \times n$ matrix A, a tolerance *eps*, and a constant integer r (e.g. 10 as in [1]), the algorithm computes an orthonormal matrix Q such that $\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\,\boldsymbol{A}\| \leq \varepsilon$ holds with probability at least $1 - \min\{m, n\} 10^{-r}$.

1.2 Task 2

Our version of the modified HOSVD algorithm is present in mHOSVD.m, it takes as input the tensor $\mathcal{Y}$, the integer $method \in \{0, 1\}$ which is 0 for the rank based with RFF, and 1 for the tolerance based approximation with ARRF, the integer *rank* which prescribes the rank R (which is the same for each dimension), the real *eps* which expresses the tolerance, an integer *oversampling* that determines the oversample for the range method (R + oversampling).
We also implemented a function testMHOSVD.m to simplify the testing of our mHOSVD function, it also takes as input a parameter *trial* that specify the amount of runs, it outputs the average relative error and average time.
The matlab file task2.m is the actual testing of the mHOSVD (both rank and tolerance based) on the function related tensor

$$f(x, y, z) = \frac{1}{\sqrt{x + y + z}}$$

on a grid $\{0.1, 0.2, \ldots, n/10\}^3$ for $n = 50$.

1.3 Task 3

We implemented our version of the HOSVD in hosvd.m, it takes as input the tensor $\mathcal{X}$, the integer $method \in \{0, 1\}$ which is 0 for the rank based version, and 1 for the tolerance based version, the integer *ranks* which prescribes the rank R (which doesn't necessarily need to equal for all dimension in our implementation), the real *eps* which expresses the tolerance.
We also implemented a function testHOSVD.m to simplify the testing of our HOSVD function, it also takes as input a parameter *trial* that specify the amount of runs, it outputs the average relative error and average time.
While in task3.m we create a random tensor of size $tensor\_size = 200$ and approximate it by Tucker decomposition with multilinear ranks (R, R, R) for $R = 5, 10, \ldots, 50$.

1.4 Routine functions

For completeness we also implemented the functions:

1. ttm.m which takes as input a tensor $\mathcal{X}$ a matrix A and an integer n, it performs the nth mode multiplication of the tensor.
2. tensormat.m that takes as input a tensor $\mathcal{X}$ an integer n, it performs the nth matricization of the tensor.

2 Proof

2.1 Task 4

We aim to derive an error bound for the modified HOSVD algorithm $\|\mathcal{X} - U_1 \circ_1 U_2 \circ_3 U_3 \circ_3 \mathcal{C}\|_F$ given that each basis matrix is obtained such that $\|(I - U_\mu U_\mu^T)X^{(\mu)}\|_F^2$

From line 5 of the algorithm in the abstract, we have that

$$\tilde{\mathcal{C}} = \tilde{U}_1^T \circ_1 \tilde{U}_2^T \circ_2 \tilde{U}_3^T \circ_3 \mathcal{X} \tag{1}$$

From line 6 of the algorithm, we have that

$$\tilde{\mathcal{C}} = V_1 \circ_1 V_2 \circ_2 V_3 \circ_3 \mathcal{C} \tag{2}$$

Thus we have

$$V_1 \circ_1 V_2 \circ_2 V_3 \circ_3 \mathcal{C} = \tilde{U}_1^T \circ_1 \tilde{U}_2^T \circ_2 \tilde{U}_3^T \circ_3 \mathcal{X} \tag{3}$$

Using the result from Exercise 8 problem set 5, we have that for any matrix U with orthonormal columns, and tensors $\mathcal{X}$ and $\mathcal{Y}$ of appropriate size, the following properties hold:

$$\mathcal{Y} = \mathcal{X} \circ_\mu U \quad \Rightarrow \quad \mathcal{X} = \mathcal{Y} \circ_\mu U^T \tag{4}$$

Therefore applying (4) on equation (3) we get

$$\mathcal{C} = V_1^T \tilde{U}_1^T \circ_1 V_2^T \tilde{U}_2^T \circ_2 V_3^T \tilde{U}_3^T \circ_3 \mathcal{X} \tag{5}$$

From line 8 of the algorithm, we have that

$$\mathcal{C} = U_1^T \circ_1 U_2^T \circ_2 U_3^T \circ_3 \mathcal{X} \tag{6}$$

Denote the orthogonal projections $\pi_\mu \mathcal{X} := U_\mu U_\mu^T \circ_\mu \mathcal{X}$

$$\tilde{\mathcal{X}} := (\pi_1 \circ \pi_2 \circ \pi_3)\, \mathcal{X} \tag{7}$$

Then we have

$$\begin{aligned}
\|\mathcal{X} - \tilde{\mathcal{X}}\|_F^2 &= \|\mathcal{X} - (\pi_1 \circ \pi_2 \circ \pi_3)\, \mathcal{X}\|_F^2 \\
&= \|\mathcal{X} - \pi_1\mathcal{X} + \pi_1\mathcal{X} - (\pi_1 \circ \pi_2 \circ \pi_3)\, \mathcal{X}\|_F^2 \\
&= \|\mathcal{X} - \pi_1\mathcal{X}\|_F^2 + \|\pi_1\mathcal{X} - (\pi_1 \circ \pi_2 \circ \pi_3)\, \mathcal{X}\|_F^2 \\
&= \|\mathcal{X} - \pi_1\mathcal{X}\|_F^2 + \|\mathcal{X} - (\pi_2 \circ \pi_3)\, \mathcal{X}\|_F^2 \\
&= \|\mathcal{X} - \pi_1\mathcal{X}\|_F^2 + \|\mathcal{X} - \pi_2\mathcal{X}\|_F^2 + \|\mathcal{X} - \pi_3\mathcal{X}\|_F^2 \\
&= \left\|\left(I - U_1U_1^T\right) X^{(1)}\right\|_F^2 + \left\|\left(I - U_2U_2^T\right) X^{(2)}\right\|_F^2 + \left\|\left(I - U_3U_3^T\right) X^{(3)}\right\|_F^2
\end{aligned} \tag{8}$$

The second equality holds by adding 0. The third equality holds thanks to Pythagoras theorem. The fourth equality holds because π_1 is orthonormal. The fifth equality holds by repeating the previous steps for π_2 and π_3. The sixth equality holds because the Frobenius norm doesn't change after matricization. Therefore from the hypothesis and (8) we have that

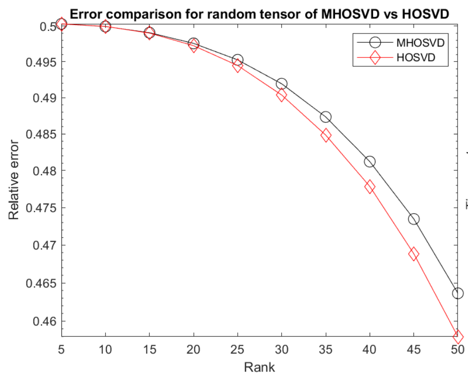
$$\|\mathcal{X} - \tilde{\mathcal{X}}\|_F^2 \leq 3\epsilon^2 \tag{9}$$

Thus we have our thesis:

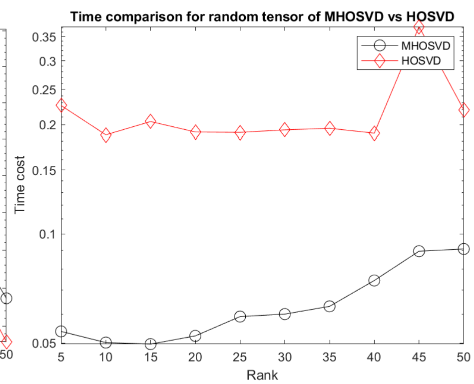
$$\|\mathcal{X} - U_1 \circ_1 U_2 \circ_3 U_3 \circ_3 \mathcal{C}\|_F = \|\mathcal{X} - \tilde{\mathcal{X}}\|_F \leq \sqrt{3}\epsilon \qquad \square \tag{10}$$

3 Results

Those are the results obtained for our analysis on task 3, with $tensor\_size = 100$ and $tensor\_size = 200$ respectively:

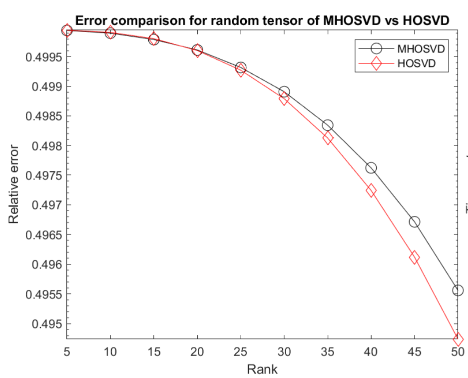


(a) Relative error comparison

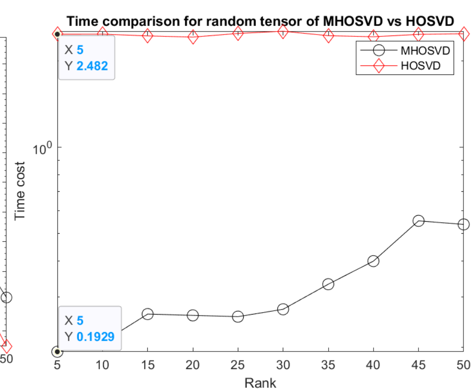


(b) Time comparison

Figure 1: Size of 100



(a) Relative error comparison



(b) Time comparison

Figure 2: Size of 200

Regarding the time comparison, our results seem to establish a clear advantage of the mHOSVD version with respect to the HOSVD version, with the difference between the two increasing with respect to $size\_tensor$. Regarding the relative error comparison, the HOSVD seems to be slightly more accurate. As sanity check for our implementation we check that relative error for both HOSVD and mHOSVD converges to the machine error when $tensor\_size = 50$.

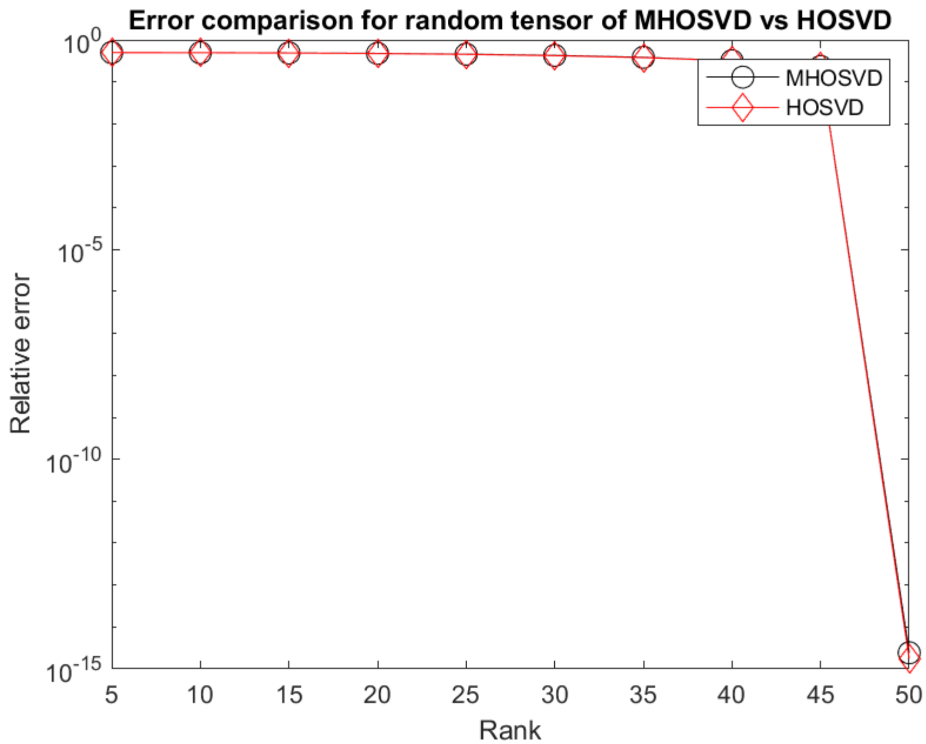


Figure 3: Relative error comparison for $tensor\_size = 50$

References

[1] Halko, N., Martinsson, P. G., Tropp, J. A. *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.* . 2011.