Multiple Tensor Times Matrix computation

Suraj Kumar

Inria & ENS Lyon

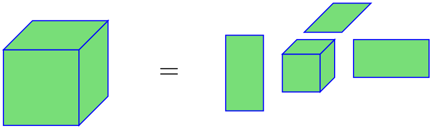
Email:*suraj.kumar@inria.fr*

CR12: October 2023

https://surakuma.github.io/courses/daamtc.html

Tucker decomposition of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$

It represents a tensor with d matrices (usually orthonormal) and a small core tensor.



Tucker decomposition of a 3-dimensional tensor.

$$\mathcal{A} = \mathcal{G} \times_1 U_1 \cdots \times_d U_d$$

$$\mathcal{A}(i_1, \cdots, i_d) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} \mathcal{G}(\alpha_1, \cdots, \alpha_d) U_1(i_1, \alpha_1) \cdots U_d(i_d, \alpha_d)$$

It can be concisely expressed as $\mathcal{A} = [\![\mathcal{G}; U_1, \cdots, U_d]\!]$.

Here r_j for $1 \leq j \leq d$ denote a set of ranks. Matrices $U_j \in \mathbb{R}^{n_j \times r_j}$ for $1 \leq j \leq d$ are usually orthonormal and known as factor matrices. The tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_d}$ is called the core tensor.

High Order SVD (HOSVD) for computing a Tucker decomposition

Algorithm 1 HOSVD method to compute a Tucker decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired rank $(r_1, \cdots, r_d)$
Ensure: $\mathcal{A} = \mathcal{G} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d$
1: **for** $k = 1$ to d **do**
2: $\quad U_k \leftarrow r_k$ leading left singular vectors of $A_{(k)}$
3: **end for**
4: $\mathcal{G} = \mathcal{A} \times_1 U_1^\mathsf{T} \times_2 U_2^\mathsf{T} \cdots \times_d U_d^\mathsf{T}$

- When $r_i < rank(A_{(i)})$ for one or more i, the decomposition is called the truncated-HOSVD (T-HOSVD)
- The collective operation $\mathcal{A} \times_1 U_1^\mathsf{T} \times_2 U_2^\mathsf{T} \cdots \times_d U_d^\mathsf{T}$ is known as Multiple Tensor-Times-Matrix (Multi-TTM) computation

Sequentially T-HOSVD (ST-HOSVD) for Tucker decomposition

- This method is more work efficient than T-HOSVD
- In each step, it reduces the size of one dimension of the tensor

Algorithm 2 ST-HOSVD method to compute a Tucker decomposition

Require: input tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, desired rank $(r_1, \cdots, r_d)$
Ensure: $[\![\mathcal{G}; U_1, \cdots, U_d]\!]$: a $(r_1, \cdots, r_d)$-rank Tucker decomposition of $\mathcal{A}$

$\mathcal{B} \leftarrow \mathcal{A}$
for $k = 1$ to d **do**
    $U_k \leftarrow r_k$ leading singular vectors of $B_{(k)}$
    $\mathcal{B} \leftarrow \mathcal{B} \times_k U_k$
end for
$\mathcal{G} = \mathcal{B}$

We can note that ST-HOSVD also performs Multi-TTM computation by doing a sequence of TTM operations, i.e, $\mathcal{G} = ((\mathcal{A} \times_1 U_1) \times_2 U_2) \cdots \times_d U_d$.

Bottleneck computations for algorithms to compute Tucker decompositions

- Multi-TTM becomes the overwhelming bottleneck computation when
  - Matrix SVD costs are reduced using randomization via sketching or
  - U_k are computed with eigen value decompositions of $B_{(k)}B_{(k)}^T$

–

Table of Contents