Multiple Tensor Times Matrix computation

Suraj Kumar

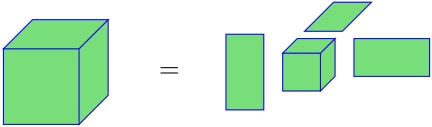
Inria & ENS Lyon
Email:*suraj.kumar@inria.fr*

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: $\mathcal{A} = \mathcal{G} \times_1 U_1 \times_2 U_2 \cdots \times_d U_d$
for $k = 1$ to $d$ do
    $U_k \leftarrow r_k$ leading left singular vectors of $A_{(k)}$
end for
$\mathcal{G} = \mathcal{A} \times_1 U_1^\mathsf{T} \times_2 U_2^\mathsf{T} \cdots \times_d U_d^\mathsf{T}$
```

- 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, $\mathcal{G} = ((\mathcal{A} \times_1 U_1) \times_2 U_2) \cdots \times_d U_d$.

Bottlenecks 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 $B_{(k)}B_{(k)}^T$

Multi-TTM computation

Let $\mathcal{Y} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ be the output tensor, $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be the input tensor, and $\mathsf{A}^{(k)} \in \mathbb{R}^{n_k \times r_k}$ be the matrix of the kth mode. Then the Multi-TTM computation can be represented as

$$\mathcal{Y} = \mathcal{X} \times_1 {\mathsf{A}^{(1)}}^{\mathsf{T}} \cdots \times_d {\mathsf{A}^{(d)}}^{\mathsf{T}}$$

$$\text{or } \mathcal{X} = \mathcal{Y} \times_1 \mathsf{A}^{(1)} \cdots \times_d \mathsf{A}^{(d)}.$$

We will focus only on the first representation in this course. Our results and analysis extend straightforwardly to the latter case.

Two approaches to perform this computation:

- TTM-in-Sequence approach – performed by a sequence of TTM operations

$$\mathcal{Y} = ((\mathcal{X} \times_1 {\mathsf{A}^{(1)}}^{\mathsf{T}}) \times_2 {\mathsf{A}^{(2)}}^{\mathsf{T}}) \cdots \times_d {\mathsf{A}^{(d)}}^{\mathsf{T}}$$

- All-at-once approach

$$\mathcal{Y}(r_1', \dots, r_d') = \sum_{\{n_k' \in [n_k]\}_{k \in [d]}} \mathcal{X}(n_1', \dots, n_d') \prod_{j \in [d]} \mathsf{A}^{(j)}(n_j', r_j')$$

$[d]$ denotes $\{1, 2, \cdots, d\}$. We represent $n_1 n_2 \cdots n_d$ and $r_1 r_2 \cdots r_d$ by n and r respectively. We mainly focus on all-at-once approach.

Final assignment – deadline Oct 24

Question: Let $\mathcal{Y} \in \mathbb{R}^{r \times r \times r \times r}$, $\mathcal{X} \in \mathbb{R}^{n \times n \times n \times n}$ and $A \in \mathbb{R}^{n \times r}$. What are the different approaches to perform the following Multi-TTM computation:

$$\mathcal{Y} = \mathcal{X} \times_1 A^\mathsf{T} \times_2 A^\mathsf{T} \times_3 A^\mathsf{T} \times_4 A^\mathsf{T}$$

Compute the exact number of arithmetic operation for each approach.

Table of Contents

1. Parallel Multi-TTM computation

Settings to compute parallel communication lower bound

- Without loss of generality, we assume that $n_1 r_1 \leq n_2 r_2 \leq \cdots \leq n_d r_d$
- The input tensor is larger than the output tensor, i.e., $n \geq r$
- The algorithm load balances the computation – each processor performs $1/P$th number of loop iterations
- One copy of data is in the system
    - There exists a processor whose input data at the start plus output data at the end must be at most $\frac{n+r+\sum_{i=1}^{d} n_i r_i}{P}$ words – will analyze amount of data transfers for this processor

Table of Contents

Optimization problems (Ballard et. al., 2023)

Lemma

Consider the following optimization problem:

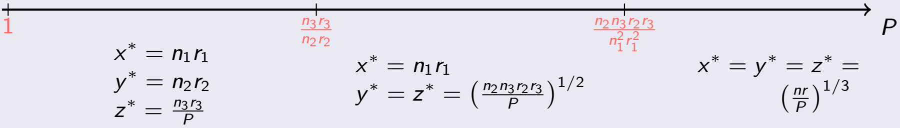
$$\min_{x,y,z} x + y + z \text{ such that}$$

$$\frac{nr}{P} \leq xyz, \quad 0 \leq \ x \ \leq n_1 r_1, \quad 0 \leq \ y \ \leq n_2 r_2, \quad 0 \leq \ z \ \leq n_3 r_3,$$

where $n_1 r_1 \leq n_2 r_2 \leq n_3 r_3$, *and* $n_1, n_2, n_3, r_1, r_2, r_3, P \geq 1$. *The optimal solution* (x^*, y^*, z^*) *depends on the relative values of the constraints, yielding three cases:*

1. *if* $P < \frac{n_3 r_3}{n_2 r_2}$, *then* $x^* = n_1 r_1$, $y^* = n_2 r_2$, $z^* = \frac{n_3 r_3}{P}$;
2. *if* $\frac{n_3 r_3}{n_2 r_2} \leq P < \frac{n_2 n_3 r_2 r_3}{n_1^2 r_1^2}$, *then* $x^* = n_1 r_1$, $y^* = z^* = \left(\frac{n_2 n_3 r_2 r_3}{P}\right)^{\frac{1}{2}}$;
3. *if* $\frac{n_2 n_3 r_2 r_3}{n_1^2 r_1^2} \leq P$, *then* $x^* = y^* = z^* = \left(\frac{nr}{P}\right)^{\frac{1}{3}}$;

which can be visualized as follows.



$x^* = n_1 r_1$
$y^* = n_2 r_2$
$z^* = \frac{n_3 r_3}{P}$

$x^* = n_1 r_1$
$y^* = z^* = \left(\frac{n_2 n_3 r_2 r_3}{P}\right)^{1/2}$

$x^* = y^* = z^* = \left(\frac{nr}{P}\right)^{1/3}$

Optimization problems (Ballard et. al., 2023)

Lemma

Consider the following optimization problem:

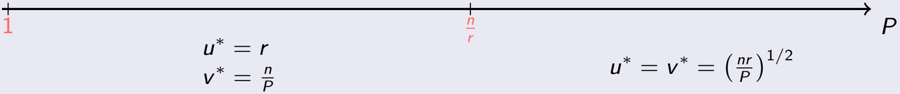
$$\min_{u,v} u + v \text{ such that}$$

$$\frac{nr}{P} \leq uv, \quad 0 \leq \; u \; \leq r, \quad 0 \leq \; v \; \leq n,$$

where $n \geq r$, *and* $n, r, P \geq 1$. *The optimal solution* (u^*, v^*) *depends on the relative values of the constraints, yielding two cases:*

1. *if* $P < \frac{n}{r}$, *then* $u^* = r$, $v^* = \frac{n}{P}$;
2. *if* $\frac{n}{r} \leq P$, *then* $u^* = v^* = \left(\frac{nr}{P}\right)^{\frac{1}{2}}$;

which can be visualized as follows.



1 $\quad \frac{n}{r}$ $\quad P$

$u^* = r$

$v^* = \frac{n}{P}$

$u^* = v^* = \left(\frac{nr}{P}\right)^{1/2}$

Both lemma can be proved using the KKT conditions.

Table of Contents