Introduction to tensors in high dimensions, and their approximation

L. Grigori

Inria Paris, UPMC

December 2020

Extreme-scale Mathematically-based Computational Chemistry

European Research Council
Established by the European Commission

informatics mathematics

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS
DEPUIS 1257

Plan

Notations

Low rank approximation algorithms

Plan

Notations

Low rank approximation algorithms

Introduction to tensors

Let $\mathcal{A}$ be a tensor of dimension d, size $n_1 \times \cdots \times n_d$, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$. A d-order tensor is the tensor product of d vector spaces.

- $d = 1$, first order tensors: vectors
- $d = 2$, second order tensors: matrices

Examples of solutions from problems in large dimensions in scientific computing.

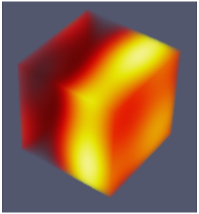


Figure: Densité de matière, projection 3d d'une densité 6d : $\int_v f(x, v)dv$

Courtesy of V. Ehrlacher and D. Lombardi

Example from scientific computing

Figure: Double stream instability for Vlasov-Poisson equation

Courtesy of V. Ehrlacher and D. Lombardi

Notations

Let $\mathcal{A}$ be a tensor of dimension d, size $n_1 \times \cdots \times n_d$, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$.

- The element of the 3rd order tensor $\mathcal{A}$ is denoted $\mathcal{A}(i_1, i_2, i_3)$
- Fibers: defined by fixing all indices except one, e.g. for 3-order tensor: column fiber $\mathcal{A}_{:i_2 i_3}$, row fiber $\mathcal{A}_{i_1 : i_3}$, tube fiber $\mathcal{A}_{i_1 i_2 :}$
- Slices: defined by fixing all indices except two, e.g. for 3-order tensor: horizontal $\mathcal{A}_{i_1 ::}$, lateral $\mathcal{A}_{: i_2 :}$, frontal $\mathcal{A}_{:: i_3}$

Presentation using notations and following [Kolda and Bader, 2009].

Operations with tensors

- Inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \dots n_d}$ is

$$(\mathcal{A}, \mathcal{B}) = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \mathcal{A}(i_1, \dots, i_d)\mathcal{B}(i_1, \dots, i_d)$$

- The norm of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots n_d}$, analogous to the Frobenius norm of a matrix, is

$$||\mathcal{A}|| = \sqrt{(\mathcal{A}, \mathcal{A})} = \sqrt{\sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \mathcal{A}^2(i_1, i_2, \dots, i_d)}$$

Rank-one tensors

A rank one tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$ is the outer product of d vectors,

$$\begin{aligned} \mathcal{A} &= u_1 \circ u_2 \circ \ldots \circ u_d, \text{that is} \\ \mathcal{A}(i_1, \ldots, i_d) &= u_1(i_1) \cdot \ldots \cdot u_d(i_d), \text{ for all } 1 \leq i_j \leq n_j, j = 1 \ldots d \end{aligned}$$

Unfoldings to transform a tensor into a matrix

- The mode-j unfolding of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots n_d}$ represents the tensor by a matrix $A_j \in \mathbb{R}^{n_j \times N}$, where $N = n_1 \cdot \ldots \cdot n_{j-1} \cdot n_{j+1} \cdot \ldots \cdot n_d$.
- Tensor element $\mathcal{A}(i_1, \dots i_d)$ is mapped to $A(i_j, k)$, where $k = 1 + \sum_{v=1, v \neq j}^{d} (i_v - 1) N$.
- Example for $\mathcal{A} \in \mathbb{R}^{3 \times 2 \times 3}$ with the frontal slices:

$$A_{::1} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \; A_{::2} = \begin{bmatrix} 7 & 10 \\ 8 & 11 \\ 9 & 12 \end{bmatrix} \; A_{::3} = \begin{bmatrix} 13 & 16 \\ 14 & 17 \\ 15 & 18 \end{bmatrix}$$

The unfoldings along modes 1, 2, and 3 are:

$$A_1 = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 \\ 2 & 5 & 8 & 11 & 14 & 17 \\ 3 & 6 & 9 & 12 & 15 & 18 \end{bmatrix} \; A_2 = \begin{bmatrix} 1 & 2 & 3 & 7 & 8 & 9 & 13 & 14 & 15 \\ 4 & 5 & 6 & 10 & 11 & 12 & 16 & 17 & 18 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \end{bmatrix}$$

Tensor multiplication along mode j with a matrix

The j-mode product of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$ with $U \in \mathbb{R}^{K \times n_j}$ is

$$\mathcal{B} = \mathcal{A} \times_j U, \ \mathcal{B} \in \mathbb{R}^{n_1 \times \ldots \times n_{j-1} \times K \times n_{j+1} \times \ldots \times n_d},$$

$$\mathcal{B}(i_1, \ldots, i_{j-1}, k, i_{j+1}, \ldots, i_d) = \sum_{i_j=1}^{n_j} \mathcal{A}(i_1, i_2, \ldots i_d) U(k, i_j)$$

By using unfoldings, this is equivalent to:

$$B_j = U A_j$$

Some properties for j-mode matrix products:

$$\mathcal{A} \times_j U_1 \times_j U_2 = \mathcal{A} \times_j (U_2 U_1)$$
$$\mathcal{A} \times_j U_1 \times_k U_2 = \mathcal{A} \times_k U_2 \times_j U_1$$

Matrix products

- The Kronecker product of $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{k \times j}$ is $C \in \mathbb{R}^{(mk)\times(nj)}$,

$$C = A \otimes B = \begin{bmatrix} A(1,1)B & \dots & A(1,n)B \\ \vdots & \ddots & \vdots \\ A(m,1)B & \dots & A(m,n)B \end{bmatrix}$$

- The Khatri-Rao product of two matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{k \times n}$ is $C \in \mathbb{R}^{(mk)\times n}$

$$C = A \odot B = [A(:,1) \otimes B(:,1) \dots A(:,n) \otimes B(:,n)]$$

- The Hadamard product of two matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{m \times n}$ is $C \in \mathbb{R}^{m \times n}$, computed as the elementwise matrix product:

$$C = A * B = \begin{bmatrix} A(1,1)B(1,1) & \dots & A(1,n)B(1,n) \\ \vdots & \ddots & \vdots \\ A(n,1)B(n,1) & \dots & A(n,n)B(n,n) \end{bmatrix}$$

Some properties of matrix products

$$
\begin{aligned}
(A \odot B)^T (A \odot B) &= A^T A * B^T B, \\
(A \odot B)^+ &= ((A^T A) * (B^T B))^+ (A \odot B)^T
\end{aligned}
$$

where A^+ denotes the Moore-Penrose pseudoinverse of A.
Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots n_d}$ and matrices $U_i \in \mathbb{R}^{m \times n_i}$ for all $1 \leq i \leq d$

$$
\begin{aligned}
\mathcal{B} &= \mathcal{A} \times_1 U_1 \times_2 U_2 \ldots \times_d U_d \\
&\Leftrightarrow \quad B_i = U_i A_i (U_d \otimes \ldots \otimes U_{i+1} \otimes U_{i-1} \otimes \ldots \otimes U_1)^T
\end{aligned}
$$

Rank of a tensor

The rank of a tensor is the number of terms in the CP decomposition that computes exactly the tensor, that is $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots n_d}$ has rank k if

$$\mathcal{A} = \sum_{i=1}^{k} u_i^1 \circ u_i^2 \circ \ldots \circ u_i^d$$

where $u_i^1 \circ u_i^2$ is the outer product of u_i^1 and u_i^2.

- Determining the rank of a tensor is an NP-problem in general [Hastad, 1990]
- Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its maximum rank is bounded as:

$$rank(\mathcal{A}) \leq \min(n_1 n_2, n_1 n_3, n_2 n_3)$$

- If the vectors a_i^1 form the columns of a matrix A^1, respectively the vectors a_i^2 and a_i^3 form A^2, A^3 and $rank(A^i) = k_i$, then the above decomposition is unique [Kruskal, 1989, Kruskal, 1977] if

$$k_1 + k_2 + k_3 \geq 2k + 2$$

Plan

Notations

Low rank approximation algorithms

Main low rank approximation algorithms

- CANDECOMP/PARAFAC (CP): tensor decomposed as a sum of rank-one tensors: proposed by Hitchcock in 1927.
- Tucker decomposition: introduced by Tucker in 1963 [Tucker, 1963], can be computed by using high order SVD (HOSVD)
- Tensor train for high dimensions

CP decomposition

Computes an approximation by factoring a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots n_d}$ into the sum of k rank-one tensors.

$$\mathcal{A} \approx \sum_{i=1}^{k} u_{1,i} \circ u_{2,i} \circ \dots \circ u_{d,i}$$

If k is given, then the CP is typically computed by the Alternating Least Squares (ALS) algorithm, that is solve

$$\min_{\mathcal{A}'} ||\mathcal{A} - \mathcal{A}'|| \text{ with } \mathcal{A}' = \sum_{i=1}^{k} \alpha_k u_{1,i} \circ u_{2,i} \circ \dots \circ u_{d,i}$$

- Let matrix $U_j \in \mathbb{R}^{n_j \times k}$ be formed by vectors $u_{j,i}$, $i = 1, \dots, k$, for all $j = 1, \dots, d$.

CP decomposition (contd)

- Let matrix $U_j \in \mathbb{R}^{n_j \times k}$ be formed by vectors $u_{j,i}$, $i = 1, \ldots, k$, for all $j = 1, \ldots, d$.

If $\mathcal{A}' = \sum_{i=1}^{k} \alpha_k u_{1,i} \circ u_{2,i} \circ \ldots \circ u_{d,i}$,
we can write in matricized form

$$\begin{aligned} {A'}_1 &= U_1(U_3 \odot U_2)^T \\ {A'}_2 &= U_2(U_3 \odot U_1)^T \\ {A'}_3 &= U_3(U_2 \odot U_1)^T \end{aligned}$$

CP decomposition (contd)

- ALS fixes $U_2, \ldots, U_d$ and minimizes for U_1, then fixes $U_1, U_3, \ldots U_d$ and minimizes for U_2, and so on until some error criterion is met.
- For $U_2, \ldots, U_d$ fixed, ALS minimizes:

$$\min_{U_1'} ||A_1 - U_1'(U_3 \odot U_2)^T||_F$$

- The optimal solution is

$$U_1' = A_1(U_3 \odot U_2)^{+T} = A_1(U_3 \odot U_2)(U_3^T U_3 * U_2^T U_2)$$

where U^+ denotes the Moore-Penrose pseudoinverse of U.
- U_i can be chosen to be random or equal to leading left singular vectors of A_i, for all $1 \leq i \leq d$.

ALS for computing a CP decomposition

Algorithm 1 ALS for computing a *rank* $-$ *k* approximation

Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, rank k

Initialize $U_i \in \mathcal{R}^{n_i \times k}$ for all $1 \leq i \leq d$

repeat

    for $i = 1$ to d **do**

        $V = U_1^T U_1 * \dots * U_{i-1}^T U_{i-1} * U_{i+1}^T U_{i+1} * \dots U_d^T U_d$, where $V \in \mathbb{R}^{k \times k}$

        $U_i = A_i (U_d \odot \dots \odot U_{i+1} \odot U_{i-1} \odot \dots \odot U_1) V^+$

        Normalize columns of U_i, store result in α

    end for

until maximum iterations reached or no further improvement obtained

Return: α, $U_1, U_2, \dots, U_d$

d-rank of a tensor

Given tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$, if for every unfolding A_i we have $k_i = rank(A_i)$, then we say $\mathcal{A}$ is a $rank - (k_1, \ldots, k_d)$ tensor.

Tucker decomposition

- Compute an approximation $\mathcal{A}'$ by decomposing a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ into the product of a core tensor $\mathcal{C}$ and matrices $U_i, i = 1, \ldots, d$.

$$\begin{aligned} \mathcal{A}' &= \mathcal{C} \times_1 U_1 \times_2 U_2 \ldots \times_d U_d \\ &= \sum_{s_1=1}^{k_1} \sum_{s_2=1}^{k_2} \ldots \sum_{s_d=1}^{k_d} \mathcal{C}(s_1, \ldots, s_d) U_1(:, s_1) \circ \ldots \circ U_d(:, s_d) \end{aligned}$$

  where $\mathcal{C} \in \mathbb{R}^{k_1 \times k_2 \times \ldots \times k_d}$, $U_i \in \mathbb{R}^{n_i \times k_i}$, $i = 1, \ldots d$.
- The matrices $U_i, i = 1, \ldots, d$ are usually orthogonal and are (or approximate) the left singular vectors of A_i, the unfolding along mode i.
- The formula to compute the value of a Tucker tensor at a given point, for $i_j = 1, \ldots n_j$ and $1 \leq j \leq d$ is:

$$\mathcal{A}(i_1, ..., i_d) = \sum_{1 \leq s_j \leq k_j, 1 \leq j \leq d} \mathcal{C}(s_1, ..., s_d) U_1(i_1, s_1) \cdots U_d(i_d, s_d)$$

HOSVD for computing a Tucker decomposition

Algorithm 2 HOSVD (High Order SVD) for computing a $rank-(k_1, k_2, \ldots k_d)$ approximation

Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, ranks $k_1, \ldots k_d$

1: **for** $i = 1$ to d **do**

2: For every unfolding A_i along mode i compute the k_i truncated SVD of A_i, $A_i = U_i \Sigma_i V_i^T$, where $U_i \in \mathbb{R}^{n_i \times k_i}$

3: **end for**

4: $\mathcal{C} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \ldots \times_d U_d^T$

Return: $\mathcal{C}$, $U_1, U_2, \ldots, U_d$

It can be used as a starting point for ALS algorithm.

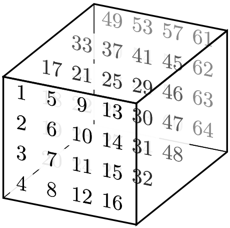
HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \rightarrow RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\mathcal{A}' = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$



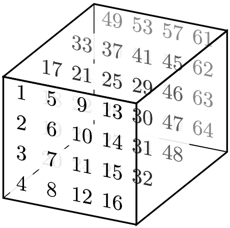
HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \to RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\mathcal{A}' = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$



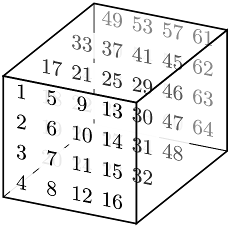
HOSVD for computing a Tucker decomposition

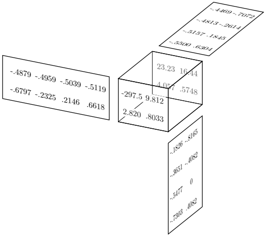
HOSVD for computing a $rank-(k_1,\dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\dots\times n_d}$, ranks $k_1,\dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i\times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \to RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\mathcal{A}' = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$





HOSVD for computing a Tucker decomposition

HOSVD for computing a $rank - (k_1, \dots k_d)$ approximation

1. **Input**: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, ranks $k_1, \dots k_d$
2. For every unfolding A_i along mode $i = 1...d$ compute the k_i (approximated) leading left singular vectors of A_i, $Q_i \in \mathbb{R}^{n_i \times k_i}$

$$A_1 = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 & 29 & 33 & 37 & 41 & 45 & 49 & 53 & 57 & 61 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 & 30 & 34 & 38 & 42 & 46 & 50 & 54 & 58 & 62 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 & 31 & 35 & 39 & 43 & 47 & 51 & 55 & 59 & 63 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & 44 & 48 & 52 & 56 & 60 & 64 \end{bmatrix} \to RRQR \begin{bmatrix} 61 & 1 \\ 62 & 2 \\ 63 & 3 \\ 64 & 4 \end{bmatrix}$$

3. $\mathcal{C} = \mathcal{A} \times_1 Q_1^T \times_2 Q_2^T \dots \times_d Q_d^T$
4. **Return:** $\mathcal{A}' = \mathcal{C} \times_1 Q_1 \dots \times_d Q_d = \mathcal{A} \times_1 Q_1 Q_1^T \dots \times_d Q_d Q_d^T$

Error bound:

If Q_i are the leading left singular vectors of unfolding A_i, then:

$$\|\mathcal{A} - \mathcal{A}'\|_F \leq \sqrt{d} \|\mathcal{A} - \mathcal{A}_{best}\|_F,$$

where $\mathcal{A}_{best}$ is the best rank-$k_1, \dots, k_d$ approximation of $\mathcal{A}$.

References (1)

Hastad, J. (1990).
Tensor rank is NP-complete.
J. of Algorithms, 11:644–654.

Kolda, T. G. and Bader, B. W. (2009).
Tensor decompositions and applications.
SIAM Review, 51(3):455–500.

Kruskal, J. B. (1977).
Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics.
Linear Algebra and its Applications, 18(2):95–138.

Kruskal, J. B. (1989).
Multiway data analysis.
chapter Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays, pages 7–18. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands.

Tucker, L. R. (1963).
Implications of factor analysis of three-way matrices for measurement of change.
In Harris, C. W., editor, *Problems in Measuring Change*, pages 122–137. University of Wisconsin Press.