Lecture1

@sayankotor

based on

23 sent 2020

Definition

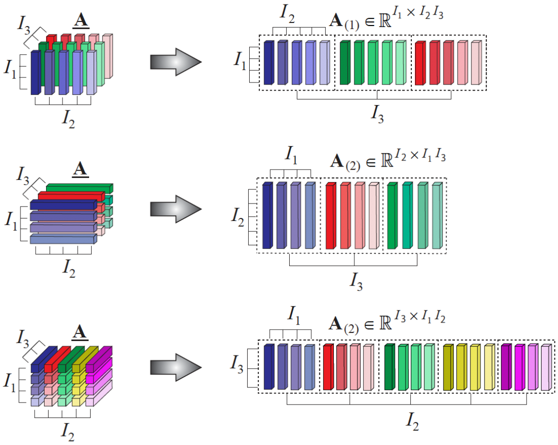
Tensor

- Tensor - multidimensional array

Casting

- Tenzor $\hat{X}$, matrix M, vector $\vec{y}$
- Strange special symbol $\hat{1}_R^T$: $\sum_{i=1}^R y_i = \hat{1}_R^T \vec{y}$
- Outer product $\circ$
- Kronecker product $\otimes$
- Hadamarad product $*$
- Khartri-rao product $\odot$
- Frobenious norm $\|\hat{X}\|_F = \sum_{i_1=1} \cdots \sum_{i_N=1} x_{i_1 \dots i_N}$

Matricization, Unfolding



Vectorization

Isomorphism that maps the element of tensor to vector:
$\mathbb{R}^{I_1 \times \cdots \times I_N} \to (I_1 \times \cdots \times I_N)$

$$j = \sum_{k=0}^{N} i_k \prod_{m=k+1}^{N} I_m \tag{1}$$

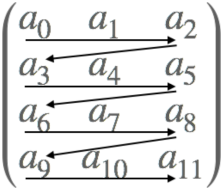


Figure: Numpy, pytorch

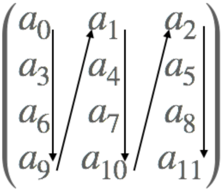


Figure: Matlab

Some vectorization properties

- $\mathrm{vec}\left(\mathbf{A}\mathbf{B}^T\right) = (\mathbf{B} \odot \mathbf{A})\mathbf{1}_R$
- $\mathrm{vec}\left(\mathbf{A}\,\mathrm{diag}(\boldsymbol{s})\mathbf{B}^T\right) = (\mathbf{B} \odot \mathbf{A})\boldsymbol{s}$
- $\mathrm{vec}\left(\mathbf{A}\mathbf{G}\mathbf{B}^T\right) = (\mathbf{B} \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{G})$
- $\mathrm{vec}(\mathbf{A} \circledast \mathbf{B}) = \mathrm{vec}(\mathbf{A}) \circledast \mathrm{vec}(\mathbf{B})$
- $\mathrm{vec}(\mathbf{A}\mathbf{B} - \mathbf{B}\mathbf{A}) = (\mathbf{I} \otimes \mathbf{A} - \mathbf{A}^T \otimes \mathbf{I})\,\mathrm{vec}(\mathbf{B})$

Tensor reshaping operations

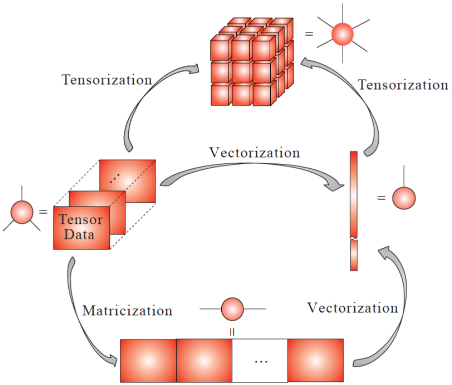


Figure: Tensor reshaping operations: Matricization, vectorization and tensorization. Matricization refers to converting a tensor into a matrix, vectorization to converting a tensor or a matrix into a vector, while tensorization refers to converting a vector, a matrix or a low-order tensor into a higher-order tensor.

Outer product

Outer product of tensors N-order tensor $\hat{A} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ and M-order tensor $\hat{A} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$ is a (N+M) order tensor $\hat{C}$:

$$c_{i_1 \ldots i_N j_1 \ldots j_M} = a_{i_1 \ldots i_N} b_{j_1 \ldots j_M} \tag{2}$$

- outer product change the number of dimensions, not dimensions!
- Outer product of $\vec{a}$, $\vec{b}$, $\vec{c}$ forms a tensor X, with entries $x_{ijk} = a_i \cdot b_j \cdot c_k$
- what is the rank of this tensor?

Mode-n product

Mode-n product of tensor $\hat{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \cdots \times I_n \times I_N}$ and matrix $B \in \mathbb{R}^{J \times I_n}$ tensor $\hat{C} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \cdots \times J \cdots \times I_N}$, that has elements:

$$c_{i_1 \ldots i_{n-1} j i_{n+1} \ldots j_N} = \sum_{i_n} a_{i_1 \ldots i_n \ldots i_N} b_{j i_n} \tag{3}$$

- n-mode product change the size of the corresponding tensor along the dimension n!
- n-mode product is the case of 3-d tensor is like matrix-matrix multiplication, where the second matrix is n-mode unfolding:

$$\hat{X} \times_n M = M \hat{X}_{[n]} \tag{4}$$

Kronecker product, left Kronecker product

For an $A \in \mathbb{R}^{I \times J}$ matrix a $B \in \mathbb{R}^{K \times L}$, the standard (Right) Kronecker product, $A \otimes_L B$, and the Left Kronecker product, $A \otimes_L B$, are the following $\mathbb{R}^{IK \times JL}$ matrices

$$C = \begin{bmatrix} a_{11}B & a_{12}B & \dots \\ \vdots & \ddots & \\ a_{J1}B & & a_{IJ}B \end{bmatrix}$$

$$C_L = \begin{bmatrix} b_{11}A & b_{12}A & \dots \\ \vdots & \ddots & \\ a_{L1}A & & b_{KL}A \end{bmatrix}$$

Kronecker product, left Kronecker product

Properties

- $(A \otimes B)^T (C \otimes D) = (A^T C) \otimes (B^T D)$
- $(A \otimes B)(E \odot F) = (AE) \odot (BF)$

Khartri-Rao product

Khartri-Rao product

- Given matrix $A \in \mathbb{R}^{I \times R}$ and $B \in \mathbb{R}^{J \times R}$
- $\mathrm{A} \odot \mathrm{B} = [\mathrm{a}_1 \otimes \mathrm{b}_1, \mathrm{a}_2 \otimes \mathrm{b}_2, \ldots, \mathrm{a}_k \otimes \mathrm{b}_k] \in \mathbb{R}^{IJ \times R}$

Properties

- $(\mathrm{A} \odot \mathrm{B})^T (\mathrm{A} \odot \mathrm{B}) = A^T A * B^T B$
- $(\mathrm{A} \odot \mathrm{B})^\dagger = ((A^T A) * (B^T B))^{-1} (\mathrm{A} \odot \mathrm{B})^T$

Hadamarad Product

Hadamarad Product

- Elementwise product of the objects with the same order and the same size

CP decomposition

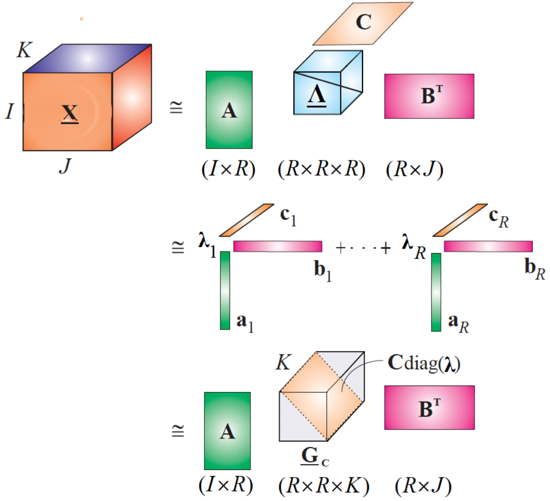
Kruskal Tensor

Kruskal tensor - tensor which be expressed as a finite sum of rank-1 tensors, in the form

$$\hat{X} = \sum_{r=1}^{R} \lambda_r \vec{b_1} \circ \vec{b_2} \cdots \circ \vec{b_n} \tag{5}$$

It also known under the names of CANDECOMP / PARAFAC, Canonical Polyadic Decomposition (CPD), or simply the CP decomposition.

CP decomposition



CP decomposizition

Tensor Rank

The tensor rank, also called the CP rank, is a natural extension of the matrix rank and is defined as a minimum number, R, of rank-1 terms in an exact CP decomposition of the form in

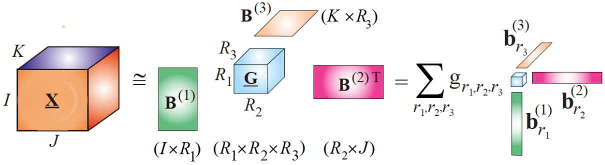
Best Decomposition

- If we define Norm $||||$ function, the best rank-r approximation is argmin $||Y - \hat{Y}||$
- For tensors of dimensions $>=$ 3 this argmin can not exists
- therefore, not every tensor has the rank
- more in https://arxiv.org/pdf/math/0607647.pdf

Tucker decomposition

$$\hat{X} = \sum_{r_1=1}^{R_1} \sum_{r_N=1}^{R_N} g_{r1,r2..rn} \vec{b_1} \circ \vec{b_2} \cdots \circ \vec{b_n} \tag{6}$$

Scheme of the Tucker decomposition



Tucker and CP decompositions

Properties of decomposition

CP	Tucker
Scalar product	
$x_{i_1,\ldots,i_N} = \sum_{r=1}^{R} \lambda_r \, b^{(1)}_{i_1,r} \cdots b^{(N)}_{i_N,r}$	$x_{i_1,\ldots,i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1,\ldots,r_N} \, b^{(1)}_{i_1,r_1} \cdots b^{(N)}_{i_N,r_N}$
Outer product	
$\underline{\mathbf{X}} = \sum_{r=1}^{R} \lambda_r \, \mathbf{b}^{(1)}_r \circ \cdots \circ \mathbf{b}^{(N)}_r$	$\underline{\mathbf{X}} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1,\ldots,r_N} \, \mathbf{b}^{(1)}_{r_1} \circ \cdots \circ \mathbf{b}^{(N)}_{r_N}$
Multilinear product	
$\underline{\mathbf{X}} = \underline{\mathbf{\Lambda}} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)}$	$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times_1 \mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times_N \mathbf{B}^{(N)}$
$\underline{\mathbf{X}} = [\![\underline{\mathbf{\Lambda}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!]$	$\underline{\mathbf{X}} = [\![\underline{\mathbf{G}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!]$
Vectorization	
$\mathrm{vec}(\underline{\mathbf{X}}) = \left(\bigodot_{n=N}^{1} \mathbf{B}^{(n)} \right) \boldsymbol{\lambda}$	$\mathrm{vec}(\underline{\mathbf{X}}) = \left(\bigotimes_{n=N}^{1} \mathbf{B}^{(n)} \right) \mathrm{vec}(\underline{\mathbf{G}})$
Matricization	
$\mathbf{X}_{(n)} = \mathbf{B}^{(n)} \mathbf{\Lambda} \left(\bigodot_{m=N,\, m \neq n}^{1} \mathbf{B}^{(m)} \right)^{\mathrm{T}}$	$\mathbf{X}_{(n)} = \mathbf{B}^{(n)} \mathbf{G}_{(n)} \left(\bigotimes_{m=N,\, m \neq n}^{1} \mathbf{B}^{(m)} \right)^{\mathrm{T}}$
$\mathbf{X}_{<n>} = (\bigodot_{m=n}^{1} \mathbf{B}^{(m)}) \mathbf{\Lambda} (\bigodot_{m=N}^{n+1} \mathbf{B}^{(m)})^{\mathrm{T}},$	$\mathbf{X}_{<n>} = (\bigotimes_{m=n}^{1} \mathbf{B}^{(m)}) \mathbf{G}_{<n>} (\bigotimes_{m=N}^{n+1} \mathbf{B}^{(m)})^{\mathrm{T}}$

Unfolding of initial tensor via factors

Very important formulas!
These formulas are used in decomposition algorithms

Canonical Decomposition

$X = [|A, B, C|]$

- $X_{[0]} = A(B \odot C)^T$
- $X_{[1]} = B(A \odot C)^T$
- $X_{[1]} = C(A \odot B)^T$

Tucker

$X = [|G; A, B, C|]$

- $X_{[0]} = A(B \times C)^T$
- $X_{[1]} = B(A \times C)^T$
- $X_{[1]} = C(A \times B)^T$

ALS

We have tensor $\hat{T}$ and want to find it's CPD decomposition $\hat{T} = [|\hat{U}^{(1)}\hat{U}^{(2)}\hat{U}^{(3)}|]$, i.e find $\hat{U}^{(1)}, \hat{U}^{(2)}, \hat{U}^{(3)}$

- We can define every factor matrix from initial tensor $\hat{T}$ and another factor matrices
- We can obtain this definition from LLS(Linear least squares)
- On the zero step we initialize factor matrices
- Every step consists of number_of_factors substep
- For every substep we update one of factor matrices in assumption that another factor matrices is fixed on this activity
- Do until given number of iteration is reached, or given accuracy is reached

LLS

- $\min_X \|Y - X\beta\|^F$
- $\min_X (Y - X\beta)^T (Y - X\beta)$
- $Y^T Y - \beta^T X^T Y - Y^T X\beta + \beta^T X^T X\beta = 0$
- $X^T Y = X^T X\beta$
- $\beta = (X^T X)^{-1} X^T Y$
- $\beta = Y^T X (X^T X)^- 1$
- $\beta = Y^T (X^T)^\dagger$, where $X^\dagger$ - pseudo-inverse (Moore–Penrose inverse matrix)

LLS

- $\|\hat{X}_{[0]} - U^{(0)}(U^{(1)} \odot U^{(2)})^T\|$
- $U^{(0)} = \hat{X}_{[0]}[(U^{(1)} \odot U^{(2)})^T]^\dagger$
- using properties, the pseudoinverse matrix can be expressed in terms of the already known objects