Tensors I: Basic Operations and Representations

Overview

Tensors: Vectors, matrices and so on …

- Definition
- Operators
- PARAFAC/Candecomp, polyadic, CP
- Tucker, HOSVD

Different Matrix Products

Kronecker product $A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}, \qquad B = \begin{pmatrix} b_1 & \cdots & b_m \end{pmatrix}$

Matrix case:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} =$$

$$= \begin{pmatrix} a_1 \otimes b_1 & a_1 \otimes b_2 & a_1 \otimes b_3 & \cdots & a_n \otimes b_{m-1} & a_n \otimes b_m \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \qquad B = \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}, \Rightarrow \quad A \otimes B = \left(\begin{array}{cc|cc} 5 & 7 & 15 & 21 \\ 6 & 8 & 18 & 24 \\ \hline 10 & 14 & 20 & 28 \\ 12 & 16 & 24 & 32 \end{array}\right)$$

Different Matrix Products

Vector case (row or column form):

$$a \otimes b = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \otimes \begin{pmatrix} b_1 & \cdots & b_m \end{pmatrix} =$$
$$= \begin{pmatrix} a_1 b & \cdots & a_n b \end{pmatrix} = \begin{pmatrix} a_1 b_1 & \cdot\cdot & a_1 b_m & \cdots & a_n b_1 & \cdot\cdot & a_n b_m \end{pmatrix}$$

$$a \otimes b = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} =$$

$$= \begin{pmatrix} a_1 b & \cdots & a_n b \end{pmatrix} = \begin{pmatrix} a_1 b_1 & \cdots & a_n b_1 \\ \vdots & & \vdots \\ a_1 b_m & \cdots & a_n b_m \end{pmatrix}$$

Different Matrix Products

Khatri-Rao product:

$$A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}, \qquad B = \begin{pmatrix} b_1 & \cdots & b_n \end{pmatrix};$$

$$A \bullet B = \begin{pmatrix} a_1 \otimes b_1 & a_2 \otimes b_2 & a_3 \otimes b_3 & \cdots & a_{n-1} \otimes b_{n-1} & a_n \otimes b_n \end{pmatrix}$$

= matching columnwise Kronecker product

only for matrices with the same number of columns!

$$A = \left(\begin{array}{c|c} 1 & 3 \\ 2 & 4 \end{array}\right), \qquad B = \left(\begin{array}{c|c} 5 & 7 \\ 6 & 8 \end{array}\right), \Rightarrow \quad A \bullet B = \left(\begin{array}{c|c} 5 & 21 \\ 6 & 24 \\ \hline 10 & 28 \\ 12 & 32 \end{array}\right)$$

Different Matrix Products

Hadamard product:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix},$$

$$A * B = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1n}b_{1n} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2n}b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & a_{m2}b_{m2} & \cdots & a_{mn}b_{nm} \end{pmatrix}$$

only for matrices of equal size!

Definition

Tensor as multi-indexed object:

One index: vector: $x = (x_i)_{i=1}^{n} = (x_{i_1})_{i_1=1}^{n} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ or $x = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}$

Two indices: matrix: $A = (A_{i,j})_{i=1,\, j=1}^{n,\quad m} = (A_{i_1,i_2})_{i_1=1, i_2=1}^{n_1,\quad n_2} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,m} \end{pmatrix}$

Three indices: cube: $A = (A_{i,j,k})_{i=1,\, j=1, k=1}^{n,\quad m \quad l} = (A_{i_1,i_2,i_3})_{i_1=1, i_2=1, i_3=1}^{n_1,\quad n_2 \quad n_3}$

$$\begin{array}{ccc} & \iddots & \\ & a_{1,1,2} & \\ a_{1,1,1} \nearrow & & \longrightarrow a_{1,2,1} \quad \cdots \\ \downarrow & & \\ a_{2,1,1} & & \\ \vdots & & \end{array}$$

Multi-index: $x = (x_{i_1 i_2 \ldots i_N})_{i_1=1, i_2=1, \ldots\ldots, i_N=1}^{n_1,\quad n_2,\quad \cdots \quad ,n_N}$

Motivation: Why tensors?

PDE for two-dimensional problems:

$$(au_x)_x + (bu_y)_y = f(x,y)$$

Discretization in 2D: $$\frac{au_{i-1,j} + au_{i+1,j} - (a+b)u_{i,j} + bu_{i,j-1} + bu_{i,j+1}}{h^2} = f_{i,j}$$

$u_{i,j}$ can be seen as a vector or as a 2-way tensor=matrix.

Linear system Au=f with block matrix A:

$$A_{ij,km} u_{km} = f_{ij}$$

So matrix $A_{ij,km}$ can be also seen as a 4-way tensor

Motivation: Why tensors?

PDE with a number of additional parameters,
high-dimensional problems:

$au_{xx} + bu_{yy} + cu_{zz} = f$ for discrete sets of parameters a_{ijk}

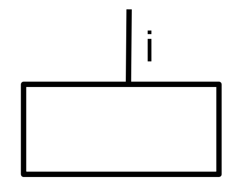
Leads to linear system A_{mn} for each i,j,k → $A_{mn,ijk}$

Classical matrix/vector problems but for huge problems:
Represent vector/matrix by tensor with efficient representation.

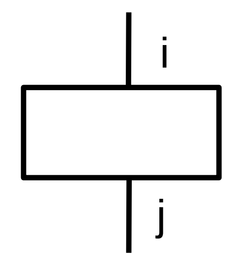
$$x_i = x_{i_1 \ldots i_N}$$

Graphical Notation

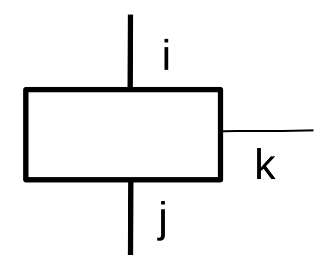
Vector (1 leg): $\left(x_i\right)_i \leftrightarrow$



Matrix (2 legs): $\left(a_{ij}\right)_{i,j} \leftrightarrow$

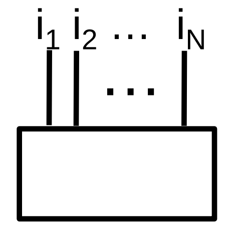


Cube (3 legs): $\left(x_{ijk}\right)_{i,j,k} \leftrightarrow$



General tensor with N legs

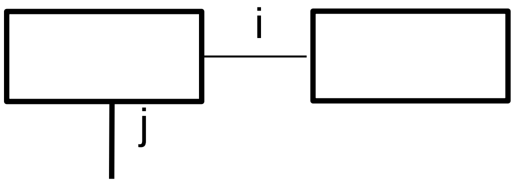
$$\left(x_{i_1 \ldots i_N}\right)_{i_1,\ldots,i_N} \leftrightarrow$$



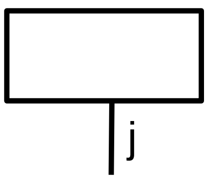
Graphical Notation

Matrix-vector product – contraction over index i:

$$\left(a_{ij}\right)_{i,j}\cdot\left(x_i\right)_i=\left(y_j\right)_j \quad \leftrightarrow$$



$$\leftrightarrow$$

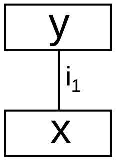


$$\sum_i a_{ij}x_i = a_{ij}x_i = y_j$$

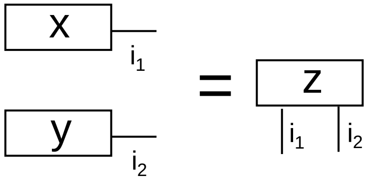
Einstein notation,
shared indices are contracted via summation.
No distinction between covariant and contravariant!

Basic Operations

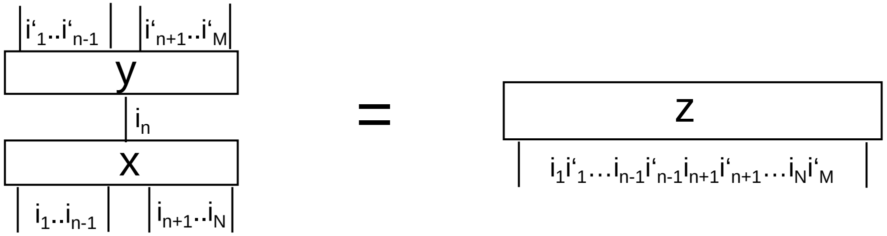
Contraction $\sum_{i_1} x_{i_1} y_{i_1}$ gives scalar z



Tensor product $x_{i_1} y_{i_2}$ gives 2-tensor $z_{i_1 i_2}$



More general: $$\sum_{i_n} x_{i_1 \cdots i_n \cdots i_N} y_{i'_1 \cdots i'_{n-1} i_n i'_{n+1} \cdots i'_M} = z_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N i'_1 \cdots i'_{n-1} i'_{n+1} \cdots i'_M}$$



Tensor as data hive of different form

$$kron(x,y) = x \otimes y = \begin{pmatrix} x_1 y & \cdots & x_n y \end{pmatrix}^T = \left(x_{i_1} y_{i_2}\right)_{i_1,i_2}$$ seen as a column vector

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_m \\ \vdots & \ddots & \vdots \\ x_n y_1 & \cdots & x_n y_m \end{pmatrix} = \left(x_{i_1} y_{i_2}\right)_{i_1,i_2}$$ seen as a matrix

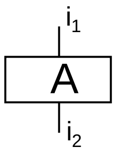
$$= kron(y^T, x) = y^T \otimes x$$

$$yx^T = \begin{pmatrix} y_1 x_1 & \cdots & y_1 x_n \\ \vdots & \ddots & \vdots \\ y_m x_1 & \cdots & y_m x_n \end{pmatrix} = \left(x_{i_1} y_{i_2}\right)_{i_1,i_2}$$ seen as a matrix

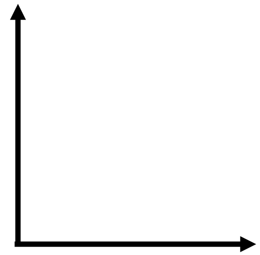
$$= kron(x^T, y) = x^T \otimes y$$

$$x \circ y = \left(x_{i_1} y_{i_2}\right)_{i_1,i_2}$$ seen as a two-leg tensor

Matrix

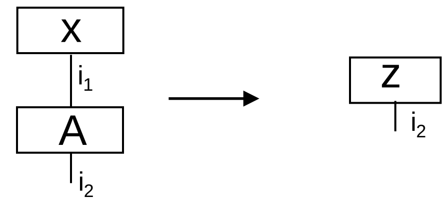
Matrix: 

$A_{i_1 i_2}$

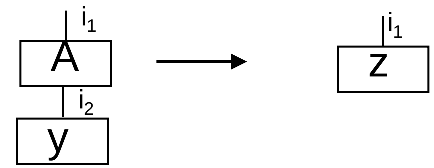


Operations: Contractions

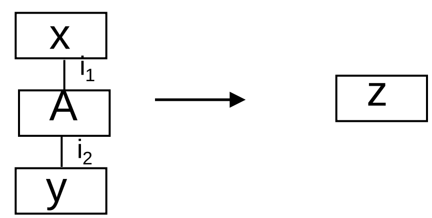
$$\sum_{i_1} A_{i_1 i_2} x_{i_1} = z_{i_2}$$



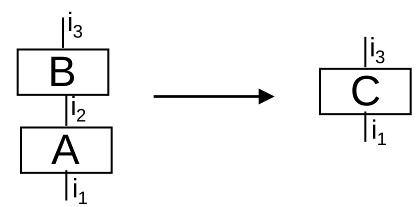
$$\sum_{i_2} A_{i_1 i_2} y_{i_2} = z_{i_1}$$



$$\sum_{i_1 i_2} A_{i_1 i_2} x_{i_1} y_{i_2} = z$$

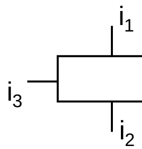


$$\sum_{i_2} A_{i_1 i_2} B_{i_2 i_3} = C_{i_1 i_3}$$

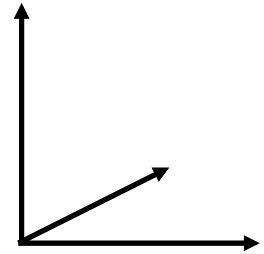


Tensor product: $A_{i_1 i_2} x_{i_3} = C_{i_1 i_2 i_3}$

Three Leg as Standardexample

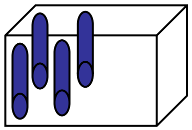


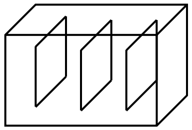
$A_{i_1 i_2 i_3}$

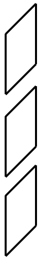


Operations: Contractions in i_1 , i_2 , i_3 or combinations gives tensor with less legs.

Tensor product gives tensor with more legs.

See tensor as - collection of vectors → fiber  $\left(a_{i_1}\right)_{i_2 i_3}$

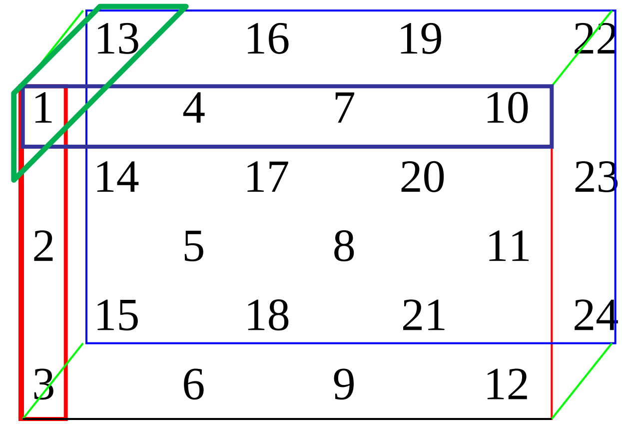
- collection of matrices → slices  $\left(A_{i_1 i_2}\right)_{i_3}$

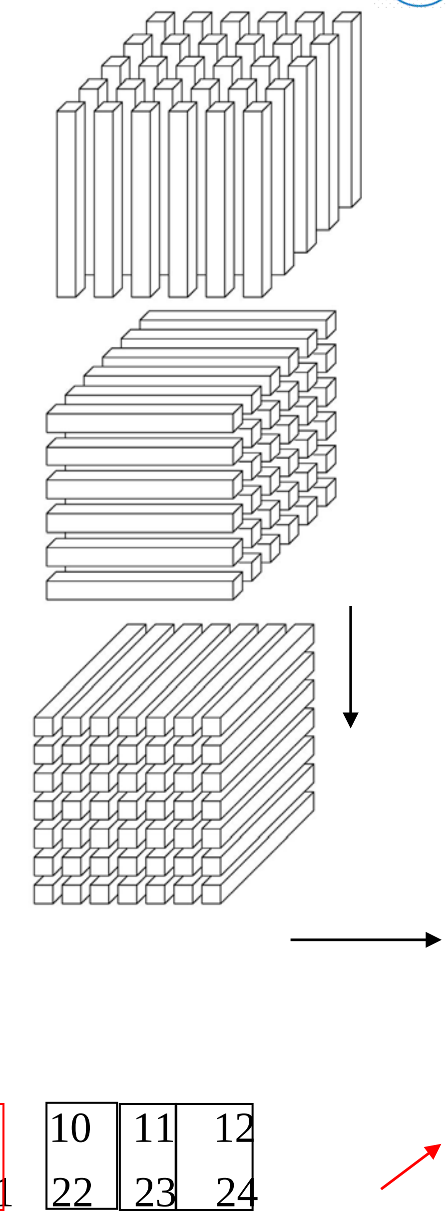
- large matrix, unfolding  $A_{\{i_1 i_2\} i_3} = A_{j_1 i_3}$

Operations between tensors are defined by contracted indices.

Fibers

A: 3 x 4 x 2 – tensor





Mode-1 fibers, $X_{:,j,k}$:

1	4	7	10	13	16	19	22
2	5	8	11	14	17	20	23
3	6	9	12	15	18	21	24

Mode-2 fibers, $X_{j,:,k}$:

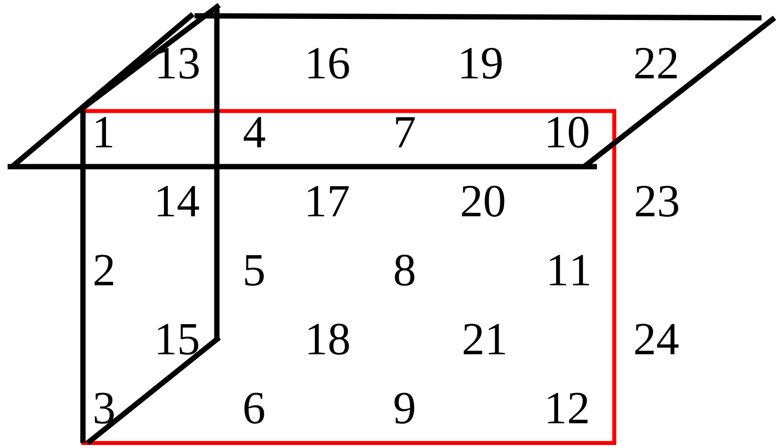
1	2	3	13	14	15
4	5	6	16	17	18
7	8	9	19	20	21
10	11	12	22	23	24

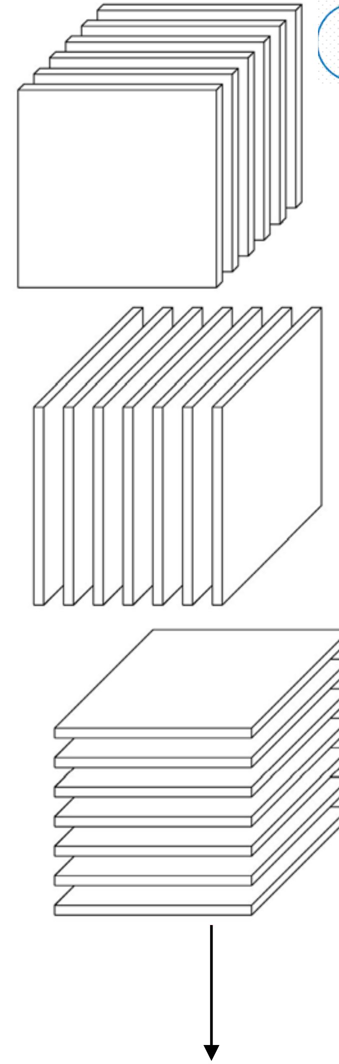
Mode-3 fibers, $X_{j,k,:}$:

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24

Slices

A: 3 x 4 x 2 – tensor





Frontal slices, 1,2: $X_{:,:,k}$

1	4	7	10
2	5	8	11
3	6	9	12

13	16	19	22
14	17	20	23
15	18	21	24

Lateral slices, 1,3: $X_{:,k,:}$

1	13
2	14
3	15

4	16
5	17
6	18

7	19
8	20
9	21

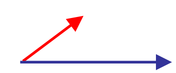
10	22
11	23
12	24

Horizontal sl. 2,3: $X_{k,:,:}$

13	16	19	22
1	4	7	10

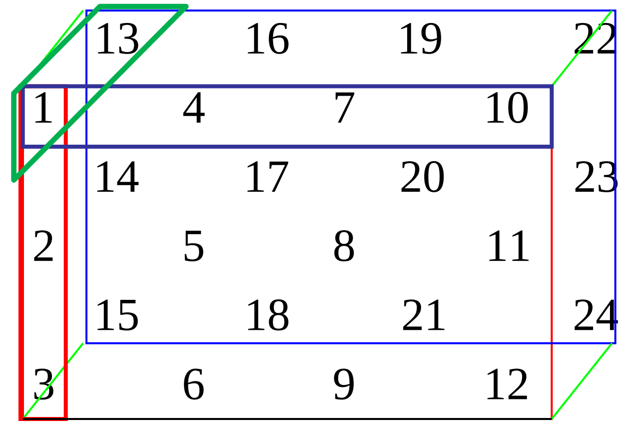
14	17	20	23
2	5	8	11

15	18	21	24
3	6	9	12



Matricification

A: 3 x 4 x 2 – tensor



Mode-1 unfolding:

$$A_{(1)} = \left(\begin{array}{cccc|cccc} \boxed{1} & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ \boxed{2} & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ \boxed{3} & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{array}\right)$$

$$A_{i_1\{i_2 i_3\}} = A_{i_1 j_1}$$

$j_1 = i_2 + n_2(i_3 - 1)$

Mode-2 unfolding

$$A_{(2)} = \begin{pmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{pmatrix}$$

$$A_{i_2\{i_1 i_3\}}$$

Mode-3 unfolding

$$A_{(3)} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{pmatrix}$$

Vectorization:

$$vec(A) = \begin{pmatrix} 1 & 2 & \cdots & 23 & 24 \end{pmatrix}^T$$

General Matricification

Tensor $\quad A_{i_1 \ldots i_n i_{n+1} \ldots i_N} \rightarrow A_{\{i_1 \ldots i_n\}\{i_{n+1} \ldots i_N\}} = A_{ij} \quad$ Matrix

$$i = i_1 + n_2(i_2 - 1) + n_2 n_3(i_3 - 1) + \ldots + n_2 \cdots n_n(i_n - 1),$$
$$j = i_{n+1} + n_{n+2}(i_{n+2} - 1) + n_{n+2} n_{n+3}(i_{n+3} - 1) + \ldots + n_{n+2} \cdots n_N(i_N - 1).$$

or with any partitioning of the indices in two groups (rows/columns)

General remark on notation:
many properties/operations with tensors are formulated using totally different notations! $\blacktriangleright, \blacktriangleleft, \odot, \otimes, \bullet, \circ, \times, \triangleright\triangleleft$

Basis Transformation

Tensor $$A = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} A_{ijk}\; e_i^{(1)} \otimes e_j^{(2)} \otimes e_k^{(3)}$$

Change of basis $$e_i^{(l)} = Q^{(l)} e'^{(l)}_i$$

$$A'_{pqr} = \left(\sum_i \sum_j \sum_k A_{ijk} Q^{(1)} e'^{(1)}_i \otimes Q^{(2)} e'^{(2)}_j \otimes Q^{(3)} e'^{(3)}_k \right)_{pqr} =$$

$$= \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} Q_{pi}^{(1)} Q_{qj}^{(2)} Q_{rk}^{(3)} A_{ijk}$$

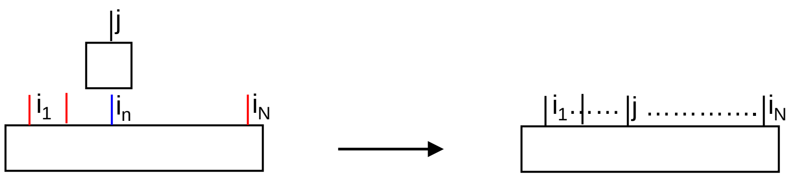
Notation: $$A' = \left(Q^{(1)}, Q^{(2)}, Q^{(3)} \right) \cdot A$$

n-Mode Product of Tensor with Matrix

Tensor Matrix

$$A_{i_1 \dots i_n \dots i_N}, \qquad U_{j i_n}: \quad \left(A \times_n U\right)_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \dots i_N} \cdot u_{j i_n} =: B_{i_1 \dots j \dots i_N}$$

Contraction over i_n, i_n replaced by index $j:=i'_n$



In the n-mode product each mode-n fiber is multiplied by the matrix U: $B_{i_1 \dots i_{n-1},:,i_{n+1} \dots i_N} = U \cdot A_{i_1 \dots i_{n-1},:,i_{n+1} \dots i_N}$

Useful relation between n-mode product and mode-n-unfolding:

$B_{(n)} = U \cdot A_{(n)}$ Unfold tensor A to matrix, multiply by U, fold back to tensor B.

Unfolding in matrix with blue and red, matrix product, back

n-Mode Products

For multiple n-mode product the order is irrelevant:

$$\boxed{n \neq m : A \times_m U \times_n V = A \times_n V \times_m U}$$

$$\sum_{i_m} \left(\sum_{i_n} A_{i_1 \ldots i_n \ldots i_m \ldots i_N} U_{j i_n} \right) V_{k i_m} =$$

$$= \sum_{i_n i_m} A_{i_1 \ldots i_n \ldots i_m \ldots i_N} U_{j i_n} V_{k i_m} = \sum_{i_m i_n} A_{i_1 \ldots i_n \ldots i_m \ldots i_N} V_{k i_m} U_{j i_n} =$$

$$= \sum_{i_n} \left(\sum_{i_m} A_{i_1 \ldots i_n \ldots i_m \ldots i_N} V_{k i_m} \right) U_{j i_n}$$

A matrix: $B = A \times_1 U \times_2 V \Leftrightarrow$

$$\left(B_{jk}\right) = \left(A_{i_1 i_2}\right) \times_1 \left(U_{j i_1}\right) \times_2 \left(V_{k i_2}\right) = U \cdot A \cdot V^T = U_{j,:} \cdot A \cdot \left(V_{k,:}\right)^T$$

especially $A \times_1 U = U \cdot A, \qquad A \times_2 V = A \cdot V^T$

n-Mode Products

For multiple n-mode product with the same n the order is relevant:

$$\boxed{A \times_n U \times_n V = A \times_n (VU)}$$

$$\sum_{i'_n} \left(\sum_{i_n} A_{i_1 \ldots i_n \ldots i_N} U_{i'_n i_n} \right) V_{ki'_n} =$$

$$= \sum_{i_n} A_{i_1 \ldots i_n \ldots i_N} \sum_{i'_n} U_{i'_n i_n} V_{ki'_n} = \sum_{i_n} A_{i_1 \ldots i_n \ldots i_N} \sum_{i'_n} V_{ki'_n} U_{i'_n i_n} =$$

$$= \sum_{i_n} A_{i_1 \ldots i_n \ldots i_N} W_{ki_n} = B_{i_1 \ldots k \ldots i_N}$$

Matrix case: $A \times_1 U \times_1 V = V \cdot U \cdot A = (VU) \cdot A,$

$$A \times_2 U \times_2 V = A \cdot U^T \cdot V^T = A \cdot (VU)^T$$

n-Mode Product with vector

n-mode vector product of tensor A with vector v:
Compute all inner products of mode-n fibers with v.

$$A \bar{\times}_n v = \left(\sum_{i_n=1}^{n_n} A_{i_1 \ldots i_n \ldots i_N} v_{i_n} \right)_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_N}$$

$$A \bar{\times}_n v \bar{\times}_m u = \left(A \bar{\times}_n v\right) \bar{\times}_{m-1} u = \left(A \bar{\times}_m u\right) \bar{\times}_n v =$$

$$= \left(\sum_{i_n=1}^{n_n} \sum_{i_m=1}^{n_m} A_{i_1 \ldots i_n \ldots i_m \ldots i_N} v_{i_n} u_{i_m} \right)_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_{m-1} i_{m+1} \ldots i_N}$$

for n<m because the order of the tensor is changed:
After contracting i_n: m → m-1

Matrix case: $A \bar{\times}_1 v = v^T \cdot A, \qquad A \bar{\times}_2 v = A \cdot v$

Properties

(1) $$(A \otimes B)(C \otimes D) = (AC) \otimes (BD),$$

(2) $$(A \otimes B)^{-T} = A^{-T} \otimes B^{-T}$$

(3) $$A \bullet B \bullet C = (A \bullet B) \bullet C = A \bullet (B \bullet C),$$

(4) $$(A \bullet B)^T (A \bullet B) = (A^T A) * (B^T B),$$

$$(A \bullet B)^{-1} = ((A^T A) * (B^T B))^{-1} (A \bullet B)^T$$

Proofs (1):

$$(A\otimes B)(C\otimes D)=$$

$$=\begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}\cdot\begin{pmatrix} c_{11}D & \cdots & c_{1k}D \\ \vdots & & \vdots \\ c_{n1}D & \cdots & c_{nk}D \end{pmatrix}=$$

$$=\begin{pmatrix} a_{11}c_{11}BD+\cdots+a_{1n}c_{n1}BD & \cdots \\ \vdots & \end{pmatrix}=$$

$$=\begin{pmatrix} (AC)_{11}BD & \cdots \\ \vdots & \end{pmatrix}=(AC)\otimes(BD),$$

Proofs (2):

$$(A \otimes B)^{-T} = A^{-T} \otimes B^{-T}$$

$$(A \otimes B)\left(A^{-1} \otimes B^{-1}\right) = \left(\left(AA^{-1}\right) \otimes \left(BB^{-1}\right)\right) = I \otimes I = I$$

$$(A \otimes B)^{T} = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}^{T} =$$

$$= \begin{pmatrix} a_{11}B^{T} & \cdots & a_{m1}B^{T} \\ \vdots & & \vdots \\ a_{1n}B^{T} & \cdots & a_{nm}B^{T} \end{pmatrix} = A^{T} \otimes B^{T}$$

Proofs (3):

$$A \bullet B \bullet C = (A \bullet B) \bullet C = A \bullet (B \bullet C),$$

$$(A \bullet B) \bullet C = \left(a_1 \otimes b_1 \quad \cdots \quad a_n \otimes b_n \right) \bullet C =$$

$$\left(\left(a_1 \otimes b_1 \right) \otimes c_1 \quad \cdots \quad \left(a_n \otimes b_n \right) \otimes c_n \right) =$$

$$= \left(a_1 \otimes b_1 \otimes c_1 \quad \cdots \quad a_n \otimes b_n \otimes c_n \right) =$$

$$A \bullet (B \bullet C),$$

because $\left(A \otimes B \right) \otimes C = A \otimes \left(B \otimes C \right) = A \otimes B \otimes C$

Proofs (4):

$$(A \bullet B)^{-1} = ((A^T A) * (B^T B))^{-1} (A \bullet B)^T$$

$$((A^T A) * (B^T B)) = (A \bullet B)^T (A \bullet B)$$

$$((A^T A) * (B^T B)) = \left(\left(a_i^T a_j \right)_{ij} * \left(b_i^T b_j \right)_{ij} \right) =$$

$$= \left(\left(a_i^T a_j \right) \left(b_i^T b_j \right) \right)_{ij} = \begin{pmatrix} \left(a_1^T a_1 \right) \left(b_1^T b_1 \right) & \cdots \\ \vdots & \end{pmatrix} =$$

$$= \begin{pmatrix} \left(a_1^T \otimes b_1^T \right) \left(a_1 \otimes b_1 \right) & \cdots \\ \vdots & \end{pmatrix} = \begin{pmatrix} a_1^T \otimes b_1^T \\ \vdots \\ a_n^T \otimes b_n^T \end{pmatrix} \left(a_1 \otimes b_1 \quad \cdots \quad a_n \otimes b_n \right) =$$

$$= \left(a_1 \otimes b_1 \quad \cdots \quad a_n \otimes b_n \right)^T \left(a_1 \otimes b_1 \quad \cdots \quad a_n \otimes b_n \right) = (A \bullet B)^T (A \bullet B)$$

n-Mode Products Tensor with Matrices

General relation between n-mode product, mode-n unfolding and Kronecker (tensor) product:

$$Y = A \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} \Leftrightarrow$$

$$Y_{(n)} = U^{(n)} \cdot A_{(n)} \cdot \left(U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)}\right)^T$$

$$Y = A \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)} = \sum_{i_1,\ldots,i_N} A_{i_1 \ldots i_N} U^{(1)}_{j_1 i_1} \cdots U^{(N)}_{j_N i_N} = B_{j_1 \ldots j_N}$$

N=2:

$$Y = A \times_1 U^{(1)} \times_2 U^{(2)} = U^{(1)} A \left(U^{(2)}\right)^T$$

$$Y_{(1)} = (U^{(1)} A (U^{(2)})^T)_{(1)} = U^{(1)} A_{(1)} (U^{(2)})^T$$

$$Y_{(2)} = (U^{(1)} A \left(U^{(2)}\right)^T)_{(2)} = (U^{(1)} A \left(U^{(2)}\right)^T)^T =$$

$$= U^{(2)} A^T \left(U^{(1)}\right)^T = U^{(2)} A_{(2)} (U^{(1)})^T$$

n-Mode Products Tensor with Matrices

$$Y_{(1)} = \left(A \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}\right)_{(1)} =$$

$$= \left(\sum_{i_1,i_2,i_3} A_{i_1 i_2 i_3} U^{(1)}_{j_1 i_1} U^{(2)}_{j_2 i_2} U^{(N)}_{j_N i_N}\right)_{(1)} =$$

$$= \left(\sum_{i_1} U^{(1)}_{j_1 i_1} \left(\sum_{i_2 i_3} A_{i_1 i_2 i_3} U^{(2)}_{j_2 i_2} U^{(3)}_{j_3 i_3}\right)\right)_{(1)} =$$

$$= \left(\sum_{i_1} B_{i_1 j_2 j_3} U^{(1)}_{j_1 i_1}\right)_{(1)} = \left(B_{i_1 j_2 j_3} \times_1 U^{(1)}_{j_1 i_1}\right)_{(1)} =$$

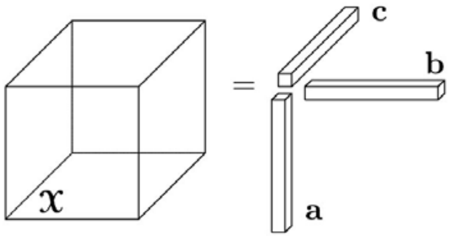
$$= U^{(1)} \left(B_{i_1 j_2 j_3}\right)_{(1)} = U^{(1)} \left(\sum_{i_2 i_3} A_{i_1 i_2 i_3} U^{(3)}_{j_3 i_3} U^{(2)}_{j_2 i_2}\right)_{(1)} =$$

$$= U^{(1)} \sum_{k} A_{i_1 k} \left(U^{(3)} \otimes U^{(2)}\right)_{r,k} = U^{(1)} A_{(1)} (U^{(3)} \otimes U^{(2)})^T$$

$$k = \{i_2 i_3\}, r = \{j_2 j_3\}$$

Rank of a tensor (3 leg case)

Rank-1 tensor:



$$\left(X_{ijk}\right) = \left(a \circ b \circ c\right)$$

3 dimensional

$$\left(X_{ijk}\right) = \left(a \otimes b \otimes c\right)$$

as vector

with vectors a, b, and c

$$X_{ijk} = a_i b_j c_k$$

Rank-R tensor for 3-leg case:

PARAFAC (parallel factors)
Candecomp (canonical decomposition)
Polyadic form
→ CP (CANDECOMP/PARAFAC)

[redacted]

$$\left(A_{ijk}\right)=\left(u_1\circ v_1\circ w_1\right)+\left(u_2\circ v_2\circ w_2\right)+\left(u_3\circ v_3\circ w_3\right)+\cdots$$

$$A_{ijk}=\sum_{r=1}^{R}\left(u_{ri}v_{rj}w_{rk}\right)$$

Tensor rank R of tensor (A_{ijk}) is the number of rank-1 terms that are necessary for representing A.

Rank representation

$$A = \sum_{r=1}^{R} u_r \circ v_r \circ w_r =$$

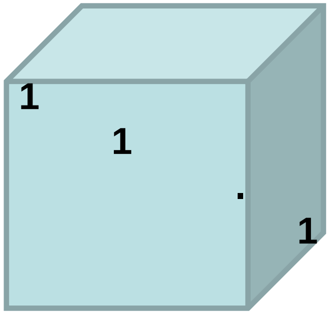
$$= \sum_{r=1}^{R} \left(\sum_{i=1}^{I} u_{ir} e_i^{(1)} \circ \sum_{j=1}^{J} v_{jr} e_j^{(2)} \circ \sum_{k=1}^{K} w_{kr} e_k^{(3)} \right) =$$

$$= \sum_{i,j,k} \left(\sum_{r=1}^{R} u_{ir} v_{jr} w_{kr} \right) e_i^{(1)} \circ e_j^{(2)} \circ e_k^{(3)}$$

With matrices U, V, and W we can write

$$A_{ijk} = \sum_{r=1}^{R} \left(u_{ir} v_{jr} w_{kr} \right) = \sum_{p,q,t} \left(u_{ip} v_{jq} w_{kt} \right) \delta_{p,q,t}$$

$$A = \left(U, V, W \right) \cdot I$$



with I the 3-way tensor with 1 on the main diagonal

U,V,W describe basis transformation with A → I

Notation

Let U, V, and W be the matrices built by the vectors u_r, v_r, and w_r. Then we can write

$$A_{(1)} = U(W \bullet V)^T,$$

$$A_{(2)} = V(W \bullet U)^T,$$

$$A_{(3)} = W(V \bullet U)^T.$$

For frontal slices $A_{(k)}$ of a three leg tensor :

$$A_{(k)} = UD^{(k)}V^T, \qquad D^{(k)} = diag(w_k)$$

Short notation: $$A = [[U,V,W]] = \sum_{k=1}^{R} u_k \circ v_k \circ w_k$$

Or more general with factor λ: $$A = [[\lambda; U,V,W]] = \sum_{k=1}^{R} \lambda_k u_k \circ v_k \circ w_k$$

Proof:

Two-leg tensor $$(u \circ v)_{(1)} = u \cdot v^T = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \cdot \begin{pmatrix} v_1 & \cdots & v_m \end{pmatrix}$$

One 3-leg tensor: $$\big(u \circ (v \circ w)\big)_{(1)} = u \cdot (w \otimes v)^T = u \cdot (w \bullet v)^T$$

General 3-leg case: $$\left(\sum_{r=1}^{R} u_r \circ v_r \circ w_r \right)_{(1)} = \sum_{r=1}^{R} \big(u_r \circ (v_r \circ w_r)\big)_{(1)} =$$

$$= \sum_{r=1}^{R} u_r \cdot (w_r \bullet v_r)^T =$$

$$= \begin{pmatrix} u_1 & \cdots & u_R \end{pmatrix} \cdot \begin{pmatrix} w_1 \bullet v_1 & \cdots & w_R \bullet v_R \end{pmatrix}^T =$$

$$= U (W \bullet V)^T$$

General N-way tensor

$$A = [[U^{(1)}, U^{(2)}, \ldots, U^{(N)}]] = \sum_{k=1}^{R} u_{1,k} \circ u_{2,k} \circ \ldots \circ u_{N,k}$$

$$A = [[\lambda; U^{(1)}, U^{(2)}, \ldots, U^{(N)}]] = \sum_{k=1}^{R} \lambda_k u_{1,k} \circ u_{2,k} \circ \ldots \circ u_{N,k}$$

Mode-n matrix formula:

$$A_{(n)} = U^{(n)} \Lambda \left(U^{(N)} \bullet \ldots \bullet U^{(n+1)} \bullet U^{(n-1)} \bullet \ldots \bullet U^{(1)} \right)^T$$

with $\Lambda = diag(\lambda)$

Proof:

3-leg tensor, proof like before:

$$\sum_r \lambda_r u_r \circ v_r \circ w_r = U\Lambda(W \bullet V)^T$$

In general:

$$U^{(1)}\Lambda\left(U^{(N)} \bullet \cdots \bullet U^{(2)}\right)^T =$$

$$= U^{(1)}\left(\lambda_1 U_1^{(N)} \otimes \cdots \otimes U_1^{(2)} \quad \cdots \quad \lambda_R U_R^{(N)} \otimes \cdots \otimes U_R^{(2)}\right)^T =$$

$$= \left(\sum_{r=1}^{R} \lambda_r U_r^{(1)} \otimes U_r^{(2)} \otimes \cdots \otimes U_r^{(N)}\right)_{(1)}$$

Low rank approximation

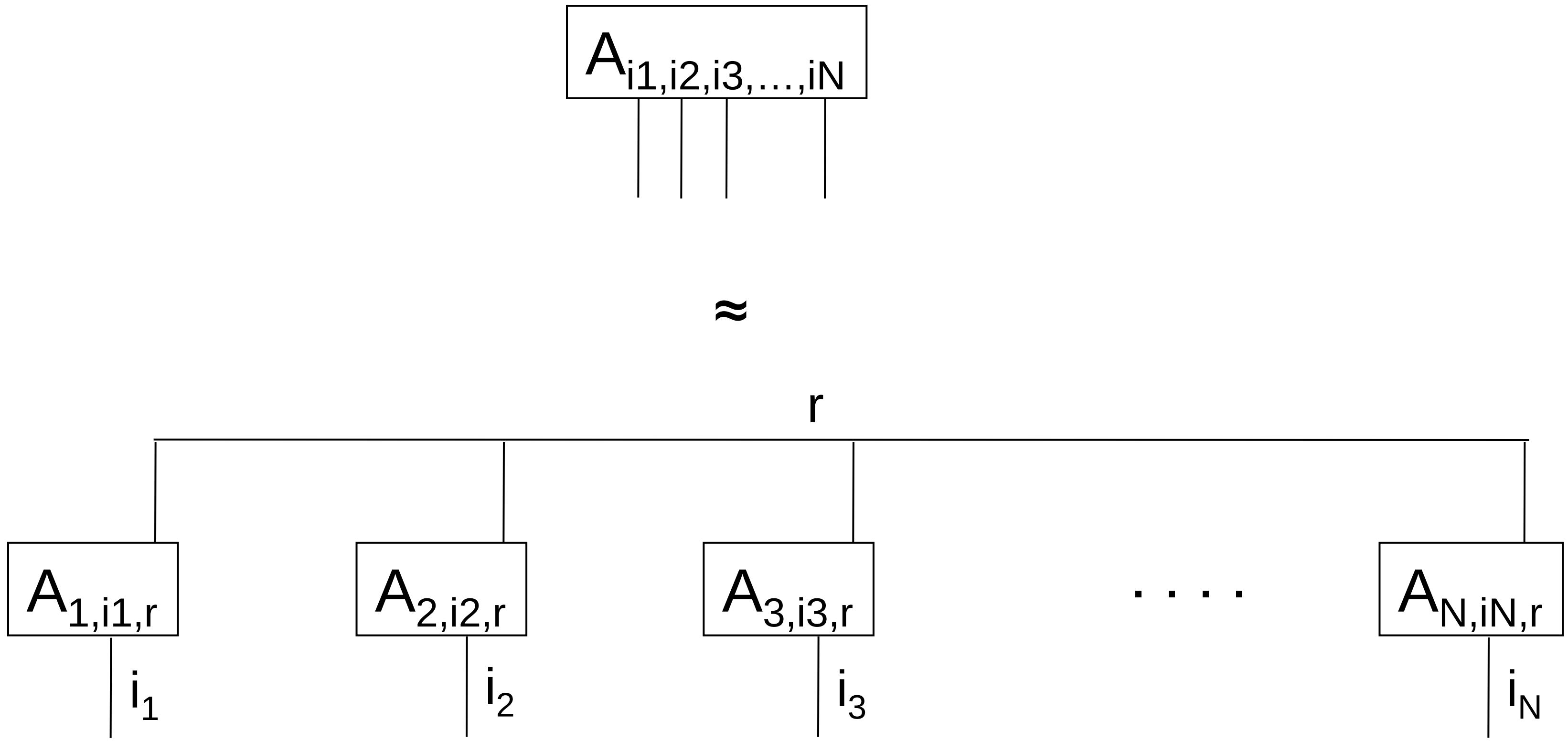
$$A_{i_1 \dots i_N} = \sum_{k=1}^{R} a_{ki_1} \dots a_{ki_N} \approx \sum_{k=1}^{r} b_{ki_1} \dots b_{ki_N}$$

(1) For R large enough every A can be represented by CP

(2) For given A there is a minimum R with this property

(3) Approximate A as good as possible by r<R

PARAFAC Graphical



$$\sum_{r=1}^{M} A_{1,i_1,r} \cdot A_{2,i_2,r} \cdot \dots \cdot A_{N,i_N,r}$$

Norm etc.

Inner product: $$\left\langle A_{i_1\ldots i_N}, B_{i_1\ldots i_N}\right\rangle = \sum_{i_1\ldots i_N=1}^{n_1,\ldots n_N} A_{i_1\ldots i_N} B_{i_1\ldots i_N}$$

Norm: $$\left\| A_{i_1\ldots i_N} \right\| = \sqrt{\sum_{i_1\ldots i_N=1}^{n_1,\ldots n_N} A_{i_1\ldots i_N}^2}$$

Rank-One tensor: $$A = a^{(1)} \circ a^{(2)} \circ \ldots \circ a^{(N)}$$ $$A_{i_1\ldots i_N} = a_{i_1}^{(1)} \cdot a_{i_2}^{(2)} \cdot \ldots \cdot a_{i_N}^{(N)}$$ with vectors $a^{(j)}$

Diagonal tensor: $$A_{i_1\ldots i_N} \neq 0 \Leftrightarrow i_1 = i_2 = \ldots = i_N$$

Symmetry

A tensor is called cubical, if every mode is of the same size, $n_1=n_2=\ldots=n_N$

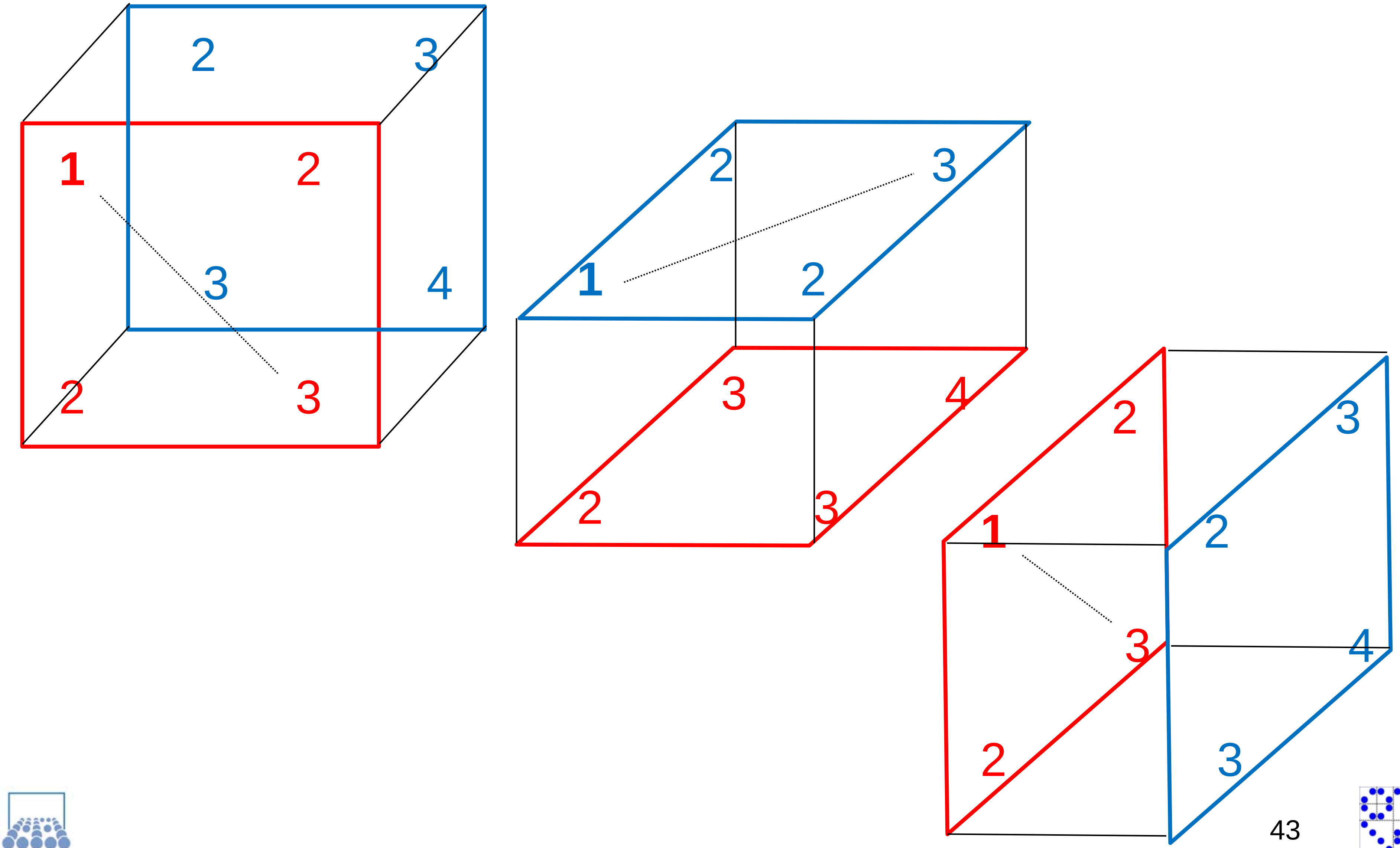
A cubical tensor is called supersymmetric, if ist elements Remain constant under any permutation of the indices:

$$A_{i_1 \ldots i_N} = A_{i_{\pi(1)} \ldots i_{\pi(N)}}$$

A tensor is partial symmetric, if it is symmetric in some modes, e.g. three-way tensor, where all frontal slices are symmetric matrices.

Example

$$A_{111} = 1, A_{112} = A_{121} = A_{211} = 2, A_{122} = A_{212} = A_{221} = 3, A_{222} = 4.$$



Results on tensor rank

$$A_{i_1 \dots i_N} = \sum_{k=1}^{R} a_{k i_1} \dots a_{k i_N}$$ with minimum R, dimension $n_1, \dots, n_N$, $n_j \le n$

General N-way tensor: R=rank $\le n^{N-1}$

Proof: Assume $n_N = n = \max n_j$.

$$A = \sum_{i_1,\dots,i_N} A_{i_1 \dots i_N} e_{i_1}^{(1)} \otimes \cdots \otimes e_{i_N}^{(N)} =$$

$$= \sum_{i_1,\dots,i_{N-1}} e_{i_1}^{(1)} \otimes \cdots \otimes e_{i_{N-1}}^{(N-1)} \otimes \left(\sum_{i_N} A_{i_1 \dots i_N} e_{i_N}^{(N)} \right)$$

Where the summation runs over maximum $\prod_{j=1}^{N-1} n_j \le n^{N-1}$ rank 1 terms.

Results on tensor rank

The true rank might be much smaller:

The maximum rank of a 3 leg tensor 3x3x3 over IR is bounded by 5.

For general 3 leg IxJxK tensor A the maximum rank is bounded by $rank(A) \le \min\{IJ, IK, JK\}$

For general 3 leg IxJx2 tensor A the maximum rank is bounded by $rank(A) \le \min\{I, J\} + \min\{I, J, \frac{\max\{I, J\}}{2}\}$

The typical rank of a 3 leg tensor 5x3x3 over IR is 5 or 6.

Results on tensor rank

Example: $A = a \otimes a + a \otimes b + b \otimes a + b \otimes b$

with linearly independent a and b, rank≤4,
with 4 linearly independent terms, but

$A = (a+b) \otimes a + b)$ with rank 1.

Theorem: rank(A)=3 for $A = v_1 \otimes v_2 \otimes w_3 + v_1 \otimes w_2 \otimes v_3 + w_1 \otimes v_2 \otimes v_3$
with linearly independent v_j, w_j.

Proof: (1) rank(A)=0 → A=0 → $v_1 \otimes a = w_1 \otimes b$!!!

(2) rank(A)=1 →

$$u \otimes v \otimes w = v_1 \otimes v_2 \otimes w_3 + v_1 \otimes w_2 \otimes v_3 + w_1 \otimes v_2 \otimes v_3$$

Assume a linear functional with

$$\varphi_1(v_1) = 1, \qquad \varphi := \varphi_1 \otimes id \otimes id$$

and apply it on above equation:

$$\varphi_1(u) v \otimes w = v_2 \otimes w_3 + w_2 \otimes v_3 + \varphi_1(w_1) v_2 \otimes v_3 =$$
$$= v_2 \otimes w_3 + (w_2 + \varphi_1(w_1) v_2) \otimes v_3$$

Left side rank 1 matrix, right side rank 2 matrix !!!

(3) Rank(A)=2:

$$u \otimes v \otimes w + u' \otimes v' \otimes w' = v_1 \otimes v_2 \otimes w_3 + v_1 \otimes w_2 \otimes v_3 + w_1 \otimes v_2 \otimes v_3$$

If u and u‘ are linearly dependent there is a functional

$$\varphi_1(u) = \varphi_1(u') = 0, \qquad \varphi_1(v_1) \neq 0 \quad or \quad \varphi_1(w_1) \neq 0$$

$$0 = \left(\varphi_1 \otimes id \otimes id\right)(A) = \varphi_1(v_1)(v_2 \otimes w_3 + w_2 \otimes v_3) + \varphi_1(w_1)v_2 \otimes v_3$$

Linearly independent !!

Hence, u and u‘ have to be linearly independent, and one of the vectors u or u‘ must be linearly independent of v_1, say u‘ is l.i. of v_1.
Choose functional with $\varphi_1(v_1) = 1, \qquad \varphi_1(u') = 0.$

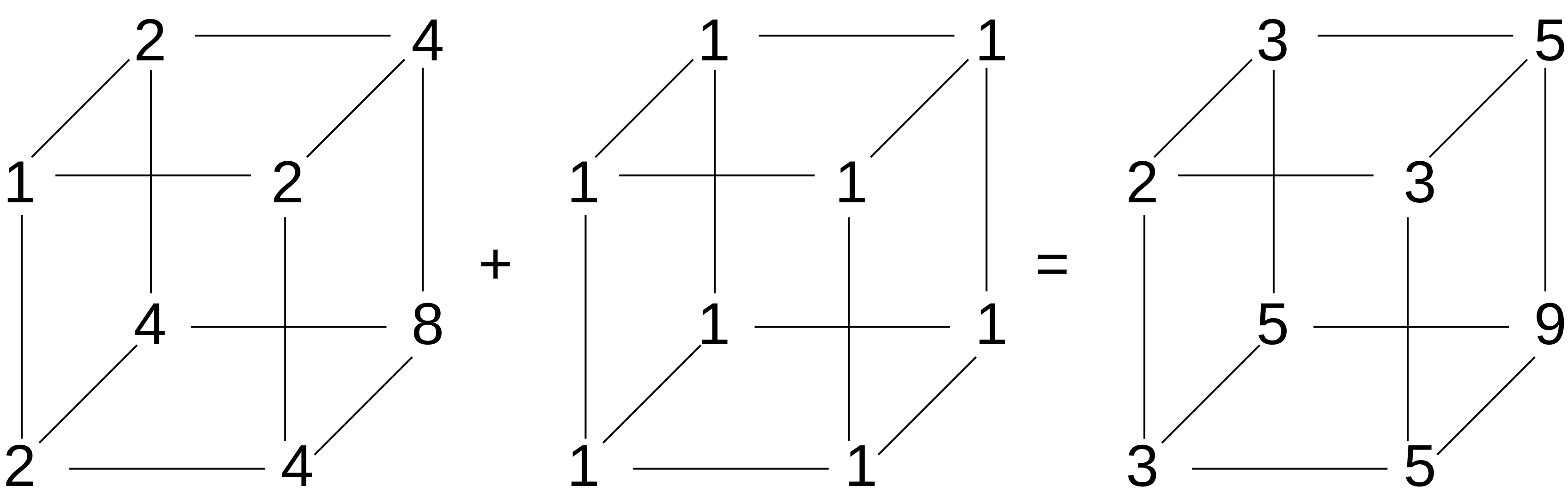
$$\varphi_1(u)v \otimes w = \left(v_2 \otimes w_3 + w_2 \otimes v_3\right) + \varphi_1(w_1)v_2 \otimes v_3$$

Again, the left-hand-side matrix is rank ≤1,
the right-hand-side matrix has rank 2 !!!

For a supersymmetric tensor we can define the symmetric rank:

$$rank_S(A) = \min\left\{ r : \quad A = \sum_{k=1}^{r} a_r \circ a_r \circ \ldots \circ a_r \right\}$$

Example: $A = (1,2)^{\otimes 3} = (1,2) \otimes \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix},$ $\qquad B = (1,1)^{\otimes 3} = (1,1) \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$



Supersymmetric of symmetric rank 2.

Rank 1:? $\quad (a,b)^{\otimes 3} = A + B,$ $\qquad$ 4 equations for 2 unknowns a,b.

$$a^3 = 2, b^3 = 9, a^2 b = 3, ab^2 = 5;$$

Smallest Typical Rank 3-way T

K			2				3		4	
J		2	3	4	5	3	4	5	4	5
	2	2	3	4	4	3	4	5	4	5
	3	3	3	4	5	5	5	5	6	6
	4	4	4	4	5	5	6	6	7	8
	5	4	5	5	5	5	6	8	8	9
	6	4	6	6	6	6	7	8	8	10
I	7	4	6	7	7	7	7	9	9	10
	8	4	6	8	8	8	8	9	10	11
	9	4	6	8	9	9	9	9	10	12
	10	4	6	8	10	9	10	10	10	12
	11	4	6	8	10	9	11	11	11	13
	12	4	6	8	10	9	12	12	12	13

DOF: R(I+J+K-2) → Expected Rank: $\left\lceil \frac{IJK}{I+J+K-2} \right\rceil$

Examples

Strassen by considering a 3-leg tensor with rank 7
Hackbusch page 69

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix}$$

with submatrices a_j, b_j, c_j

$$c_\nu = \sum_{\mu,\lambda=1}^{4} t_{\nu,\mu,\lambda} a_\mu b_\lambda$$

t is of rank 7.

Matrix case: SVD

For a tensor that is a vector, the rank is 1.

For a tensor that is a nxm matrix, the rank is given by the singular value decomposition

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i \left(u_i v_i^T\right) = \sum_{i=1}^{r} \sigma_i \left(u_i \otimes v_i\right)$$

r = the number of nonzero singular values.

For low rank approximation we can delete the small singular values.

Uniqueness of CP

Matrix case: A nxm matrix of rank r:

$$A = U_{n,r} V_{r,m}^T = \sum_{k=1}^{r} u_r \circ v_r$$

Every matrix factorization of this form gives a CP representation.

QR-factorizations, SVD.

In the matrix case (2-leg-case) the rank representations are not unique!

Uniqueness 3 leg case

Let A be a three-way tensor of rank R:

$$A = [[U, V, W]] = \sum_{k=1}^{R} u_k \circ v_k \circ w_k$$

Uniqueness is related to other rank R representations upto scaling and upto permutations

$A = [[U, V, W]] = [[U\Pi, V\Pi, W\Pi]]$ for any RxR permutation Π

$A = \sum_{k=1}^{R} (\alpha_k u_k) \circ (\beta_k v_k) \circ (\gamma_k w_k)$ with $\alpha_k \beta_k \gamma_k = 1$, for k=1,…,R

k-rank of a matrix

The k-rank of a matrix A - denoted by k_A – is the maximum number k such that any k columns of A are linearly independent.

$$T = [[A, B, C]]$$

Then the CP representation of A is unique if

$$k_A + k_B + k_C \geq 2R + 2$$

T an IxJxK-Tensor:

Then the CP representation of T is unique if

$$\min\{I, R\} + \min\{J, R\} + \min\{K, R\} \geq 2R + 2$$

For R≤K the CP representation of T is unique if

$$2R(R-1) \leq I(I-1)J(J-1)$$

The CP representation is unique for an N-way rank R tensor

$$A = [[A^{(1)}, A^{(2)}, \dots, A^{(N)}]] = \sum_{k=1}^{R} a_k^{(1)} \circ a_k^{(2)} \circ \cdots \circ a_k^{(N)}$$

if $\sum_{n=1}^{N} k_{A^{(n)}} \geq 2R + (N-1)$

Approximation of tensor by CP

Matrix case trivial via SVD: keep larger singular values and replace smaller one by 0.

For 3-way tensors this is not so easy. Especially for

$$A = \sum_{k=1}^{R} \lambda_k u_k \circ v_k \circ w_k$$

summing up r of these terms will not give a good rank-r approximation.

For finding the best rank-r approximation we have to determine all factors simultaneously!

Rank-r approximation

The situation is even worse: the best rank-r approximation might even not exist!

Consider $A = u_1 \circ v_1 \circ w_2 + u_1 \circ v_2 \circ w_1 + u_2 \circ v_1 \circ w_1$

where the matrices U, V, and W have linearly independent Columns.

Approximation by rank-2 tensors:

$$B_\alpha = \alpha \left(u_1 + \frac{1}{\alpha} u_2 \right) \circ \left(v_1 + \frac{1}{\alpha} v_2 \right) \circ \left(w_1 + \frac{1}{\alpha} w_2 \right) - \alpha \left(u_1 \circ v_1 \circ w_1 \right)$$

$$\left\| A - B_\alpha \right\| = \frac{1}{\alpha} \left\| u_2 \circ v_2 \circ w_1 + u_2 \circ v_1 \circ w_2 + u_1 \circ v_2 \circ w_2 + \frac{1}{\alpha} u_2 \circ v_2 \circ w_2 \right\| \xrightarrow{\alpha \to \infty} 0$$

Example for degeneracy!

Another example:

$$A(n) = n^2\left(x+\frac{1}{n^2}y+\frac{1}{n}z\right)^{\otimes 3} + n^2\left(x+\frac{1}{n^2}y-\frac{1}{n}z\right)^{\otimes 3} - 2n^2 x^{\otimes 3}$$

with linearly independent x,y,z.

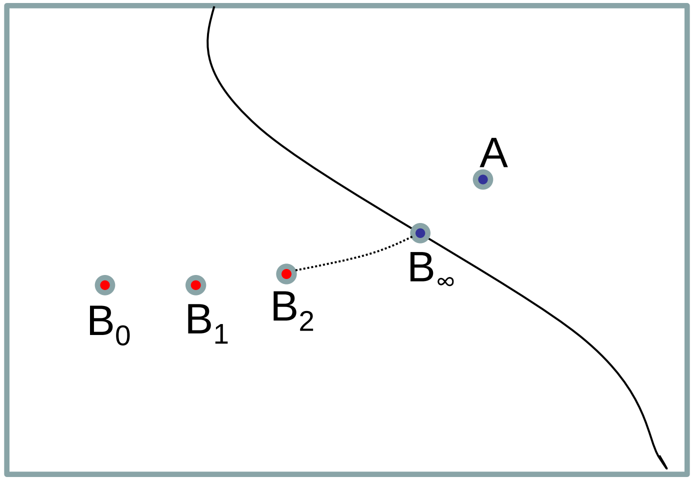
The sequence of rank 3 tensors converges for n→∞ to the rank 5 tensor:

$$A(\infty) = x\otimes x\otimes y + x\otimes y\otimes x + y\otimes x\otimes x + x\otimes z\otimes z + z\otimes x\otimes z + z\otimes z\otimes x$$

Rank spaces

Hence a sequence of rank-2 tensors converges against a rank-3 tensor:
The space of rank-2 tensors is not closed!

We can approximate the 3-way tensor as good as we want by rank-2 tensors, but the sequence of approximations does not converge in the rank-2 space.



Computing the CP

Standard method: Alternating Least Squares method (ALS)

Given any (high-rank) tensor A
Compute r-rank approximation in tensor B

$$\min_B \|A - B\| \quad \text{with} \quad B = \sum_{k=1}^{r} \lambda_k u_k \circ v_k \circ w_k = [[\lambda; U, V, W]]$$

ALS approach: fix two matrices, e.g. V and W, and solve for U.
This leads to the matrix minimization

$$\min_{\hat{U}} \left\| A_{(1)} - \hat{U}(W \bullet V)^T \right\|_F$$

with solution $\hat{U} = A_{(1)}\left((W \bullet V)^T\right)^{-1} = A_{(1)}(W \bullet V)\left(W^T W * V^T V\right)^{-1}$

Computing

Advantage: Compute pseudoinverse of small rxr-matrix

Afterwards, λ is defined by normalization

$$\lambda_k = \|\hat{u}_k\|, \qquad u_k = \hat{u}_k / \lambda_k, \qquad k = 1,\dots,r$$

In this way we update U, then V, then W, then again U and so on until convergence.

Costs per step:

ELS

ALS with enhanced line search

Assume, ALS has computed new U_{new} replacing U_{old}. Hence, we have a change in the direction $\Delta = U_{new} - U_{old}$ in the form $U_{new} = U_{old} + \Delta$.

We generalize this by introducing line search and step size μ in the form

$$U_{new} = U_{old} + \mu\,\Delta$$

looking for an optimal value of μ.

$$\min_{\mu} \left\| A - \sum_{k=1}^{R} \left(u_k + \mu\delta_k\right) \circ v_k \circ w_k \right\|^2$$

$$= \min_{\mu} \left\| \left(A - \sum_{k=1}^{R} u_k \circ v_k \circ w_k\right) - \mu \sum_{k=1}^{R} \delta_k \circ v_k \circ w_k \right\|^2$$

$$= \min_{\mu} \left\| B - \mu C \right\|^2 \rightarrow \mu \rightarrow U_{new} = U_{old} + \mu\Delta$$

ELS general

$U_{new} = U_{old} + \mu\, \Delta_U$, $V_{new} = V_{old} + \mu\, \Delta_V$, $W_{new} = W_{old} + \mu\, \Delta_W$,

$$\min_{\mu} \left\| A - \sum_{k=1}^{R} \left(u_k + \mu \delta_{u,k}\right) \circ \left(v_k + \mu \delta_{v,k}\right) \circ \left(w_k + \mu \delta_{w,k}\right) \right\|^2$$

$$= \min_{\mu} \left\| B - \mu^3 C - \mu^2 D - \mu E \right\|^2$$

$$= \min_{\mu} a_0 + a_1 \mu + a_2 \mu^2 + a_3 \mu^3 + a_4 \mu^4 + a_5 \mu^5 + a_6 \mu^6$$

Find the 5 roots of the derivative and choose the root with minimum value of the objective function.
Gives new U, V, and W.
Use ALS for new search directions and repeat.

Application of the CP

Starting point: 3-leg tensors often have small rank and the low-rank approximation is unique.

Therefore, the best approximating rank-1 term can give useful information on the data:
- Mixtures of analytes can be separated
- Concentrations can be measured
- Pure spectra and profiles can be estimated

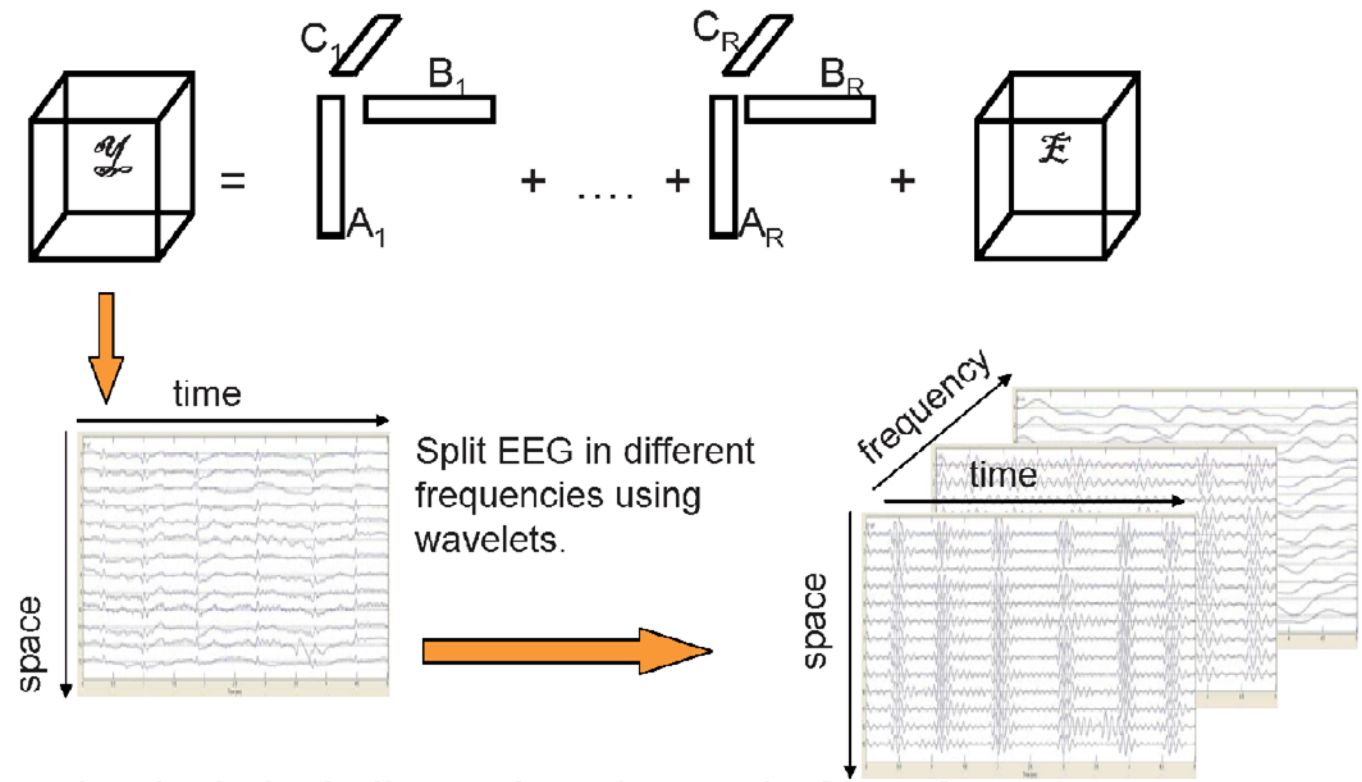
Typical example: 3-way data in time, space frequency

Translate matrix case by additional index in 3-leg tensor to achieve uniqueness!

Application of the CP

Van Huffel: PARAFAC in EEG monitoring

EEG data as 3-way tensor



=> Analysis in 3 dimensions instead of just 2

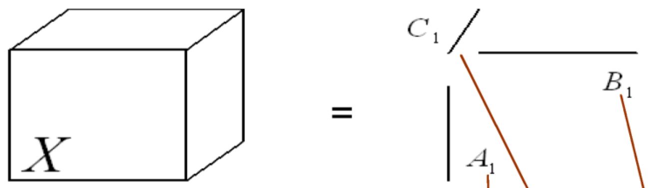
EEG Monitoring

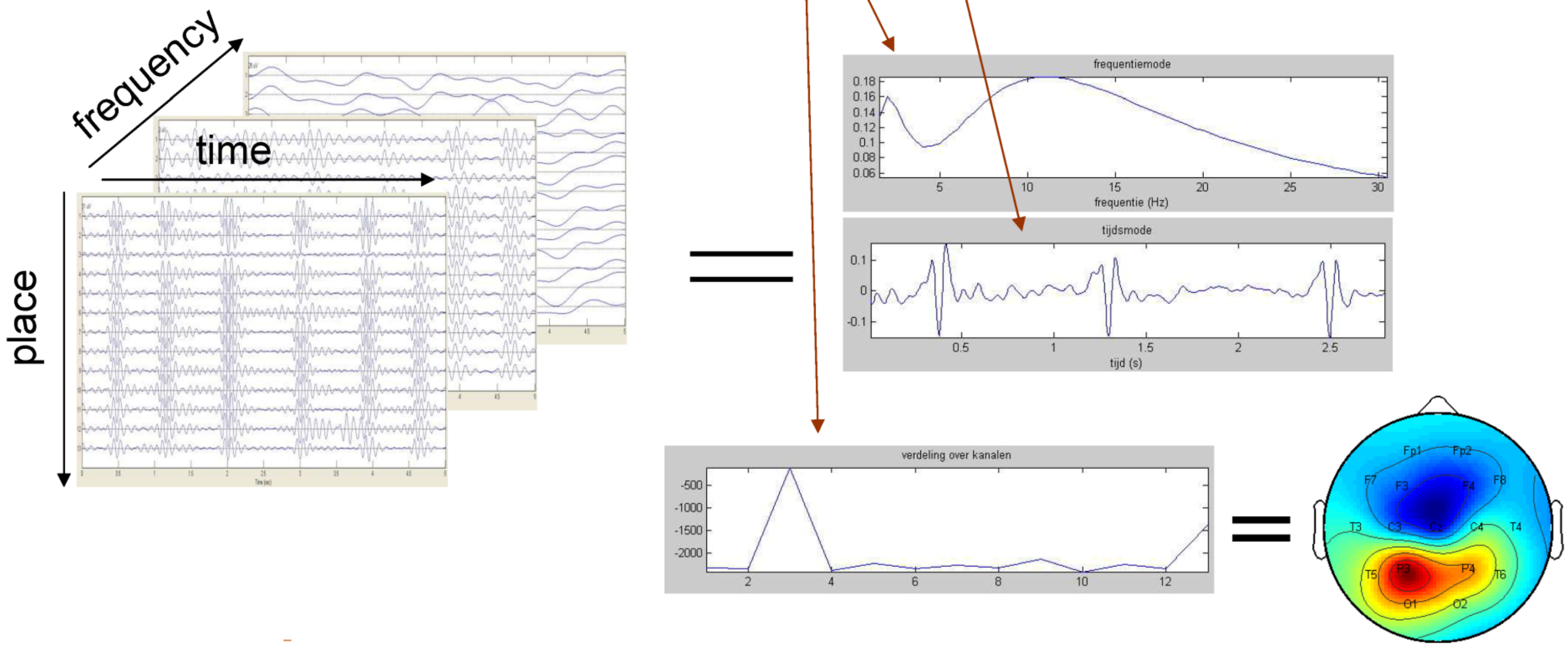
PARAFAC: Example extracting 1 component

B_1: time course

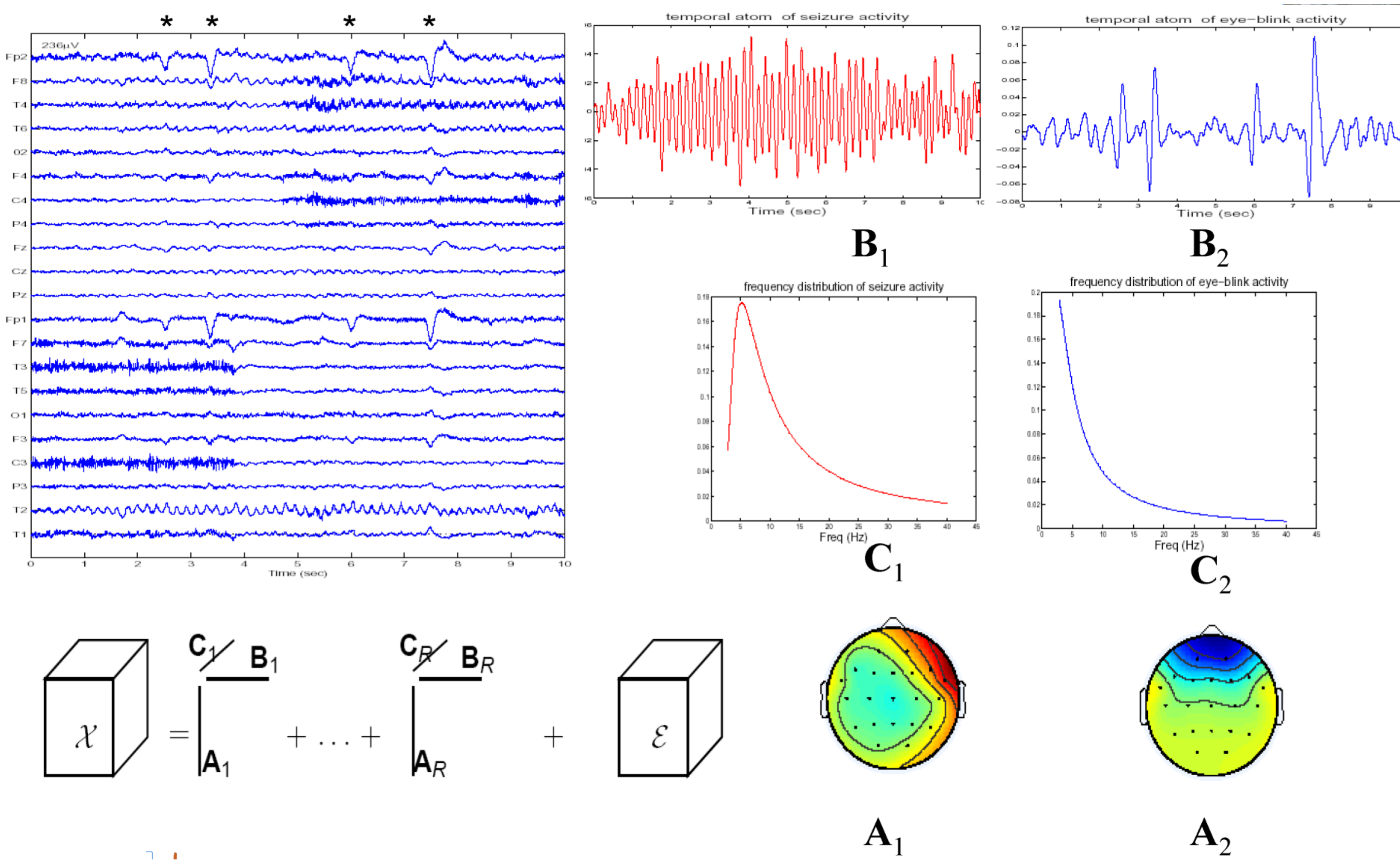
A_1: distribution over channels

C_1: frequency content (distribution across scales).



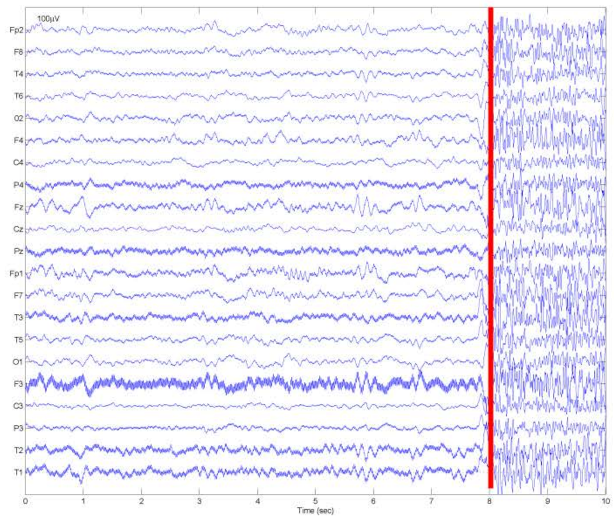


EEG rank terms

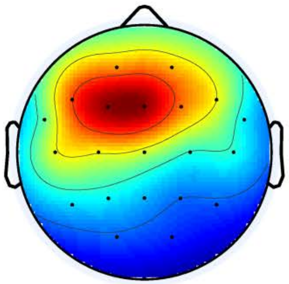


EEG: epileptic seizure onset localization

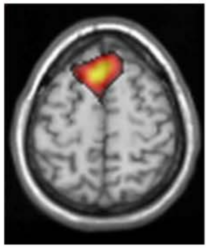
A



B

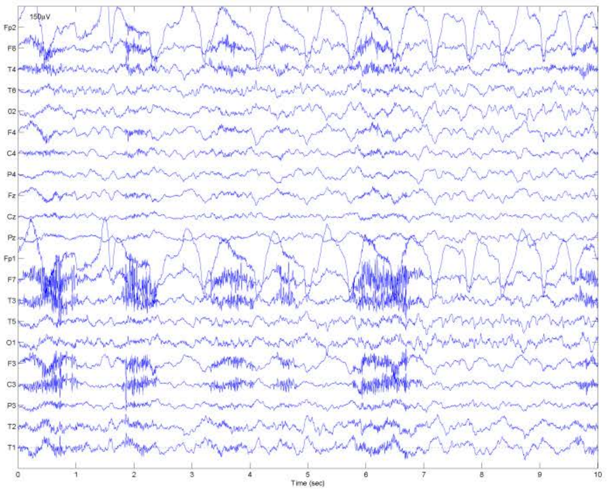


C

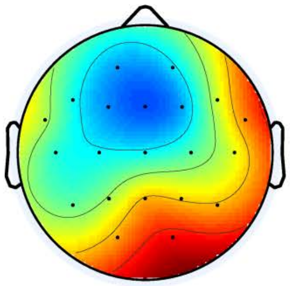


EEG

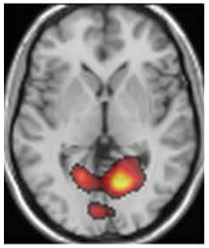
A



B



C



Better localization by CP than visually or by other matrix techniques.

Block PARAFAC (L,L,1)

Consider more general higher rank terms (L,L,1)
Because larger blocks might be necessary for accurate representation of the data.

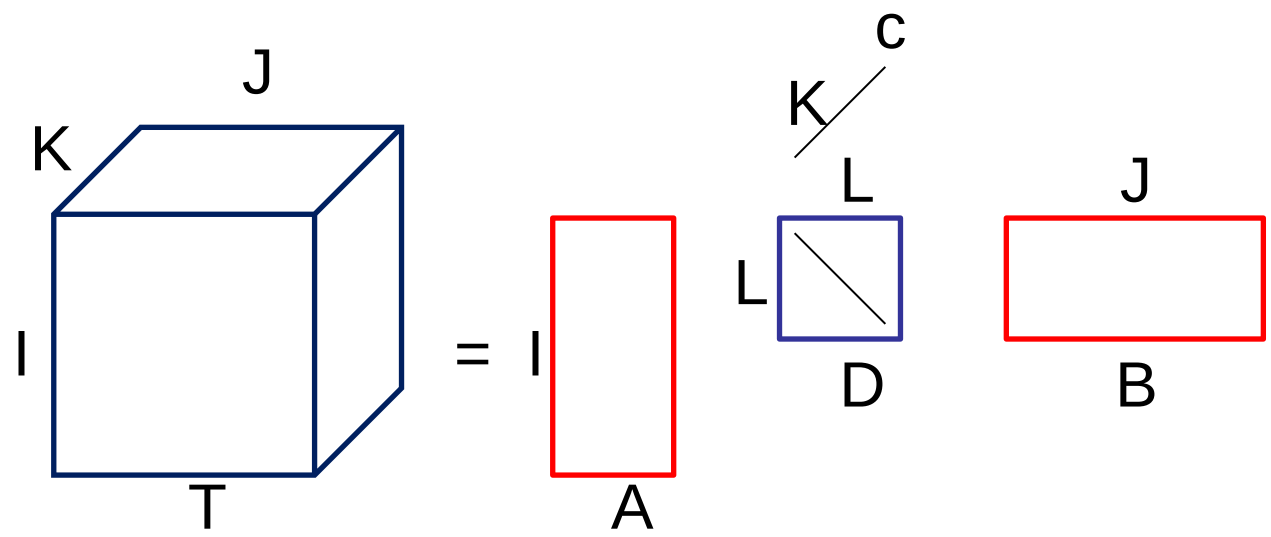
$$T = \sum_{r=1}^{R} E_r \otimes c_r, \qquad E_r : I \times J - matrix, \qquad rank(E_r) = L$$

$$T = \sum_{r=1}^{R} \left(A_r \cdot B_r^T\right) \otimes c_r$$

Also block representations are often unique, e.g.
for RL ≤ min(I,J) and C without proportional colmuns.

„Essentially unique“, upto - permutations,
- factor between A and B
- scaling

Visualization



$$T = (A, B, c) \cdot D = [[D; A, B, c]] = \sum_{r=1}^{R} D_r \times_1 A_r \times_2 B_r \times_3 c_r$$

Waring Problem

Write given integer n in the form $n = n_1^d + \cdots + n_{k_d}^d$

Proved by Hilbert 1909.
What is the minimum number k_d?

Reformulation in polynomials:
Which is the minimum s such that a degree d polynomial can be written as a sum of powers of d of linear terms:

$$P = L_1^d + \cdots + L_s^d$$

Answered by Hirschowitz 1995.

Symmetric Rank of Polynomial

The minimum s is called the symmetric rank of the polynomial P

Reformulation:
Consider map

$$\begin{aligned} \nu_d : IP(S_1) &\to IP(S_d) \\ L \quad &\to L^d \end{aligned}$$

Image of this map is called d-th Veronese variety $X_{n,d}$

Veronese Variety for Tensors

$$\nu_d : IP(V) \to IP(S^d V) \subset IP(V^{\otimes d})$$
$$v \qquad \to v^{\otimes d}$$

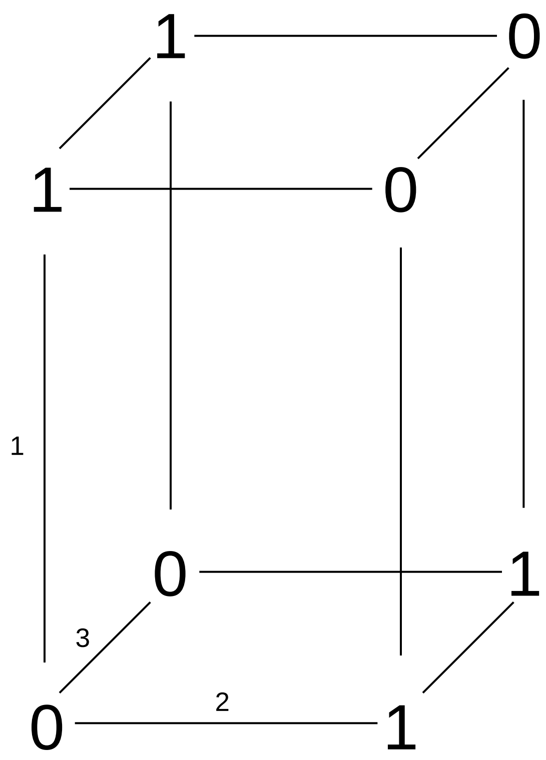
The image of this map is the Veronese Variety of IP(V)

The symmetric rank of a symmetric tensor $T \in S^d(V)$ is the minimum s with

$$T = v_1^{\otimes d} + \cdots + v_s^{\otimes d}$$

Mode n-Rank of a Tensor

View the tensor as collection of vectors in the n-th index (fibers)
The rank of these collection of vectors is the mode n-rank.



Example with $R_1=R_2=2$, $R_3=1$

Mode n=3:
Vectors (0,0), (1,1), (1,1), (0,0)

$$R_3 = rank\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} = 1$$

Mode n-rank is the rank of the mode-n unfolding matrix $A_{(n)}$

Tucker Decomposition

(three-mode) factor analysis (Tucker, 1966)
N-mode PCA (principal component analysis)
Higher-order SVD (HOSVD) (De Lathauwer, 2000)
N-mode SVD

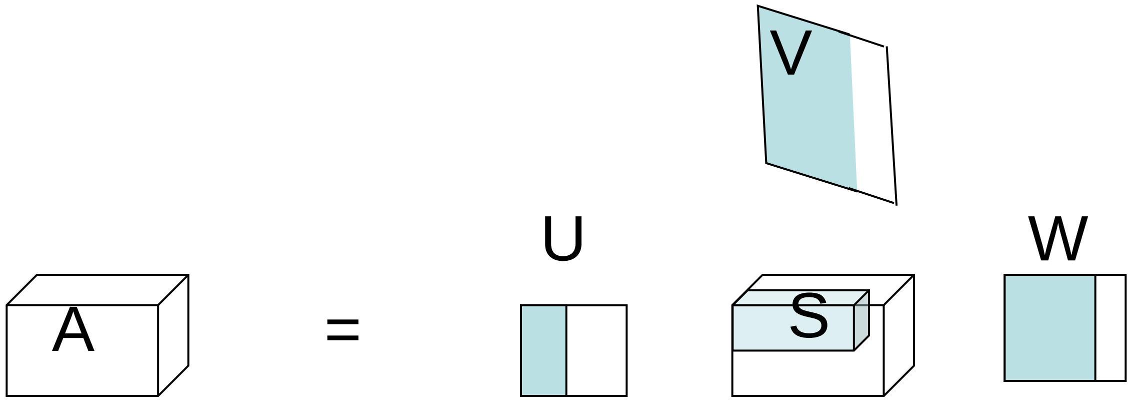
Idea: decompose given N-way tensor into a core N-way tensor with less entries in each dimension multiplied by a matrix along each mode.

$$A = G \times_1 U \times_2 V \times_3 W = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{k=1}^{K} g_{pqk} u_p \circ v_q \circ w_k =$$

$$= [[G; U, V, W]] = \left(U, V, W\right) \cdot G$$

With core tensor G and U,V,W matrices relative to each mode

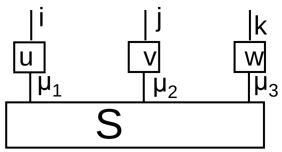
Tucker Decomposition



Unfolding in i and {jk}, SVD → U
Backfolding ΛV
Repeat for other unfoldings.

$$\left(A_{ijk}\right) = \left(S_{ijk}\right) \times_1 U^{(1)} \times_2 V^{(1)} \times_3 W^{(1)}$$

$$\left(A_{ijk}\right) = \sum_{\mu_1=1}^{k_1} \sum_{\mu_2=1}^{k_2} \sum_{\mu_3=1}^{k_3} S_{\mu_1\mu_2\mu_3} \cdot u_{\mu_1}^{(1)} \circ v_{\mu_2}^{(1)} \circ w_{\mu_3}^{(1)}$$



$$A_{ijk} = \sum_{\mu_1}^{k_1} \sum_{\mu_2}^{k_2} \sum_{\mu_3}^{k_3} S_{\mu_1\mu_2\mu_3} u_{\mu_1 i} v_{\mu_2 j} w_{\mu_3 k}, i = 1,\ldots,I, \quad j = 1,\ldots,J, \quad k = 1,\ldots,K$$

Multilinear rank (k_1,k_2,k_3)

Computation

$$A_{(1)} : I_1 \times I_2 I_3; \qquad SVD: \quad A_{(1)} = U^{(1)} \Sigma^{(1)} V^{(1)^T}$$

$$A_{(2)} : I_2 \times I_1 I_3; \qquad SVD: \quad A_{(2)} = U^{(2)} \Sigma^{(2)} V^{(2)^T}$$

$$A_{(3)} : I_3 \times I_1 I_2; \qquad SVD: \quad A_{(3)} = U^{(3)} \Sigma^{(3)} V^{(3)^T}$$

$$S = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 U^{(3)^T}$$

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

U with orthonormal columns
S all-orthogonal and ordered

Proof:

$$n \neq m: A \times_m U \times_n V = A \times_n V \times_m U$$

$$A \times_n U \times_n V = A \times_n (VU)$$

$$B = A \times_n U \Leftrightarrow \quad B_{(n)} = U \cdot A_{(n)}$$

$$S = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 U^{(3)^T}$$

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} =$$

$$= \left(A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 U^{(3)^T} \right) \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} =$$

$$= A \times_1 \left(U^{(1)^T} U^{(1)} \right) \times_2 \left(U^{(2)^T} U^{(2)} \right) \times_3 \left(U^{(3)^T} U^{(3)} \right) =$$

$$= A \times_1 I \times_2 I \times_3 I = A$$

Core Tensor S all-orthogonal:

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

with the additional property

$$\left\langle S_{i,:,:}, S_{j,:,:} \right\rangle = 0 \qquad for \quad i \neq j$$

Proof:

$$S_{(1)} = U^{(1)} A_{(1)} \left(U^{(3)} \otimes U^{(2)} \right)$$

$$S_{(1),i} = U_i^{(1)} A_{(1)} \left(U^{(3)} \otimes U^{(2)} \right)$$

$$\left\langle S_{(1),i}, S_{(1)j} \right\rangle = \left(U^{(3)} \otimes U^{(2)} \right)^T A_{(1)}^T \left(U_i^{(1)^T} U_j^{(1)} \right) A_{(1)} \left(U^{(3)} \otimes U^{(2)} \right)$$

and similarly for index 2 and 3.

Properties

Mode-n singular values = norms of slices = sing.v. of A_n

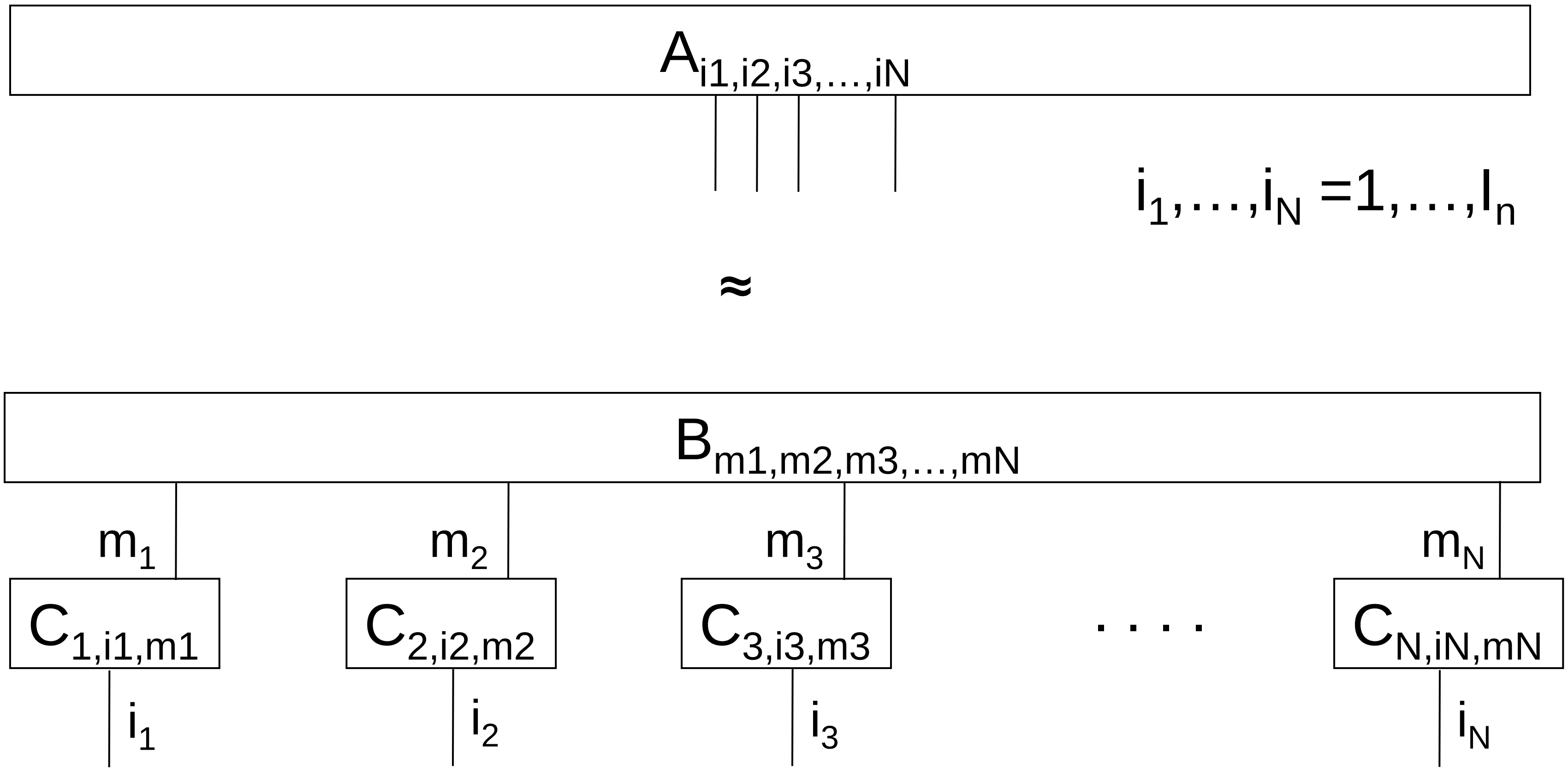
Truncate by deleting small singular values/vectors

$$S = A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 U^{(3)^T} \rightarrow$$

$$\tilde{S} = A \times_1 \tilde{U}^{(1)^T} \times_2 \tilde{U}^{(2)^T} \times_3 \tilde{U}^{(3)^T}$$

$$A \rightarrow \tilde{A} = \tilde{S} \times_1 \tilde{U}^{(1)} \times_2 \tilde{U}^{(2)} \times_3 \tilde{U}^{(3)}$$

Tucker Graphical



$$\sum_{m_1,\ldots,m_N}^{D_{small}} B_{m_1,\ldots,m_N} C_{m_1 i_1} \cdots\cdots C_{m_N i_N} ,$$

Three-way Tucker

$$A = G \times_1 U \times_2 V \times_3 W = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{k=1}^{K} G_{pqk} u_p \circ v_q \circ w_k = [[G; U, V, W]]$$

$$A_{ijm} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{k=1}^{K} G_{pqk} u_{ip} v_{jq} w_{mk},$$

$$A_{(1)} = U \cdot G_{(1)} (W \otimes V)^T$$

$$A_{(2)} = V \cdot G_{(2)} (W \otimes U)^T$$

$$A_{(3)} = W \cdot G_{(3)} (V \otimes U)^T$$

N-way Tucker

$$A = G \times_1 U^{(1)} \times_2 U^{(2)} \ldots \times_N U^{(N)} = [[G; U^{(1)}, U^{(2)}, \ldots, U^{(N)}]]$$

$$A_{i_1 i_2 \ldots i_N} = \sum_{k_1=1}^{R_1} \sum_{k_2=1}^{R_2} \cdots \sum_{k_N=1}^{R_N} G_{k_1 k_2 \ldots k_N} u^{(1)}_{i_1 k_1} u^{(2)}_{i_2 k_2} \ldots u^{(N)}_{i_N k_N}, \quad i_n = 1, \ldots, I_n$$

$$A_{(n)} = U^{(n)} \cdot G_{(n)} \left(U^{(N)} \otimes \ldots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \ldots \otimes U(1) \right)^T$$

so Tucker1 is decomposition relative to only one index,
Tucker2 relative to 2 indices, and
Tucker relative to all indices.

Computing the Tucker Dec.

For n=1,…,N

$U^{(n)}$:= matrix of left singular vectors of $A_{(n)}$

$$G := A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 \cdots \times_N U^{(N)^T}$$

Output: G, $U^{(1)}$, …, $U^{(N)}$.

We can use this algorithm also for approximating A by Choosing in $U^{(n)}$ only the dominant left singular vectors!

$R_n \rightarrow r_n$

Approximating Tucker dec.

$$\min_{G,U^{(1)},\ldots,U^{(N)}} \left\| A - [[G;U^{(1)},\ldots,U^{(N)}]] \right\|$$

$$subject\ to \quad G \in IR^{r_1\times\ldots\times r_N}, \quad U^{(n)} \in IR^{I_n\times r_n}\ columnwise\ orthogonal$$

Rewrite as minimizing $\left\| vec(A) - \left(U^{(N)} \otimes \cdots \otimes U^{(1)}\right) vec(G) \right\|$

with solution $G := A \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times_3 \cdots \times_N U^{(N)^T}$

$$\left\| A - [[G;U^{(1)},\ldots,U^{(N)}]] \right\|^2$$

$$= \left\|A\right\|^2 - 2\left\langle A, [[G;U^{(1)},\ldots,U^{(N)}]] \right\rangle + \left\| [[G;U^{(1)},\ldots,U^{(N)}]] \right\|^2$$

$$= \left\|A\right\|^2 - 2\left\langle A \times_1 U^{(1)^T} \times_2 \cdots \times_N U^{(N)^T}, G \right\rangle + \left\|G\right\|^2$$

$$= \left\|A\right\|^2 - 2\left\langle G, G \right\rangle + \left\|G\right\|^2 = \left\|A\right\|^2 - \left\|G\right\|^2$$

$$= \left\|A\right\|^2 - \left\| A \times_1 U^{(1)^T} \times_2 \cdots \times_N U^{(N)^T} \right\|^2$$

ALS for Tucker

$$\max_{U^{(n)}} \left\| A \times_1 U^{(1)^T} \times_2 \cdots \times_N U^{(N)^T} \right\|$$

$$\textit{subject to } U^{(n)} \in IR^{I_n \times r_n} \textit{ columnwise orthogonal}$$

$$\max_{U^{(n)}} \left\| U^{(n)^T} W \right\| \textit{ with } W = A_{(n)} \left(U^{(N)} \otimes \cdots \otimes U^{(n+1)} \otimes U^{(n-1)} \otimes \cdots \otimes U^{(1)} \right)$$

ALS method:

For n=1,…,N:

choose $U^{(n)}$ the r_n dominant singular vectors of W

Repeat until convergence

Uniqueness

Tucker is not unique:

$$[[G; A, B, C]] = [[G \times_1 U \times_2 V \times_3 W; AU^{-1}, BV^{-1}, CW^{-1}]]$$

Application: Tensorfaces

Given a database of images of different persons, e.g. with different looks=expressions, illumination, positions=views.
We can collect all the images in a big 5-leg tensor

$$A = \left(a_{i_{people}, i_{views}, i_{illum}, i_{express}, i_{pixel}}\right)$$

In the example there is a database of 28 male persons With 5 poses, 3 illuminations, 3 expressions, each image 512x352 pixels. Hence, A is a 28x5x3x3x7943 tensor.

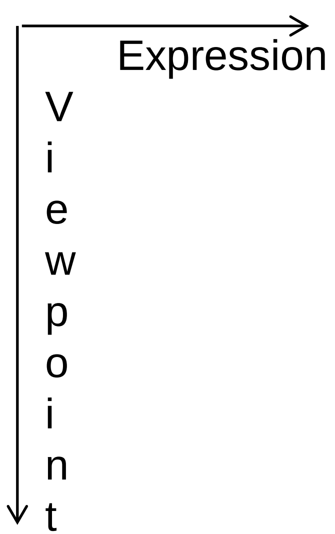
Database

28 subjects with 45 images per person.

Expression: smile

Illuminations 1 2 3

45 images for one person:



Principal Component Analysis PCA

Use eigenfaces to capture the important features in a compact form. Often eigendecomposition and eigenvectors are used in PCA.

Here we use the Tucker decomposition:

$$A = Z \times_1 U_{people} \times_2 U_{views} \times_3 U_{illum} \times_4 U_{express} \times_5 U_{pixels}$$

resulting in

$$\underbrace{A_{(pixels)}}_{\text{image data}} = \underbrace{U_{pixels}}_{\text{basis vectors}} \underbrace{Z_{(pixels)}\left(U_{express} \otimes U_{illum} \otimes U_{views} \otimes U_{people}\right)^T}_{\text{coefficients}}$$

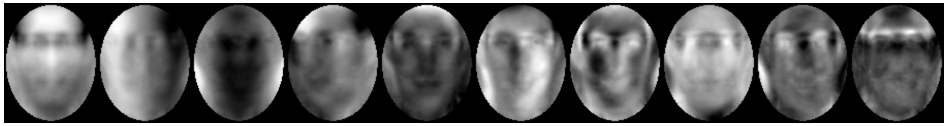
Interpretation

$$\underbrace{A_{(pixels)}}_{\text{image data}} = \underbrace{U_{pixels}}_{\text{basis vectors}} \; \underbrace{Z_{(pixels)}\left(U_{express} \otimes U_{illum} \otimes U_{views} \otimes U_{people}\right)^T}_{\text{coefficients}}$$

The mode matrix U_{pixels} can be interpreted as PCA.

By the core tensor Z we can transform the eigenimages present in U_{pixels} into eigenmodes representing the principal axes of variation across the various factors (people, viewpoints, illuminations, expressions) by forming $Z \times_5 U_{pixels}$

Eigenfaces



The first 10 PCA eigenvectors (eigenfaces) contained in the mode matrix U_{pixels}

„Multilinear Analysis of Image Ensembles: TensorFaces" by M.A.O. Vasilescu and D. Terzopoulos

Similar paper on PCA on human motion via Tensors.

PCA-CP-Tucker

PCA - bilinear model: $x_{ij} = \sum_f a_{if} b_{jf} + e_{ij}, \qquad i = 1,\ldots, I;\ j = 1,\ldots, J;$

$$X = AB^T + E$$

CP - trilinear model: $x_{ijk} = \sum_f a_{if} b_{jf} c_{kf} + e_{ijk},$

$$X_k = AD_k B^T + E_k, \qquad D_k = diag(C(k,:));$$

Tucker3: $X = AG(C \otimes B)^T + E$

with unitary bases A, B, C for each mode.