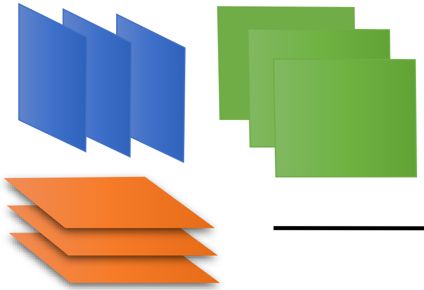
UNIVERSIDADE
FEDERAL DO CEARÁ



Overview of tensor decompositions and applications to wireless communications

André L. F. de Almeida
Federal University of Ceara, Brazil

Workshop on Low-Rank Approximations and
their Interactions with Neural Networks

LoRAINNE'24

26 November 2024

Disclaimer

A bit of many things..

Some history facts

- From the 60s: Tensor decompositions used for analyzing collections of data matrices viewed as three-way arrays:
  - 1966: Tucker decomposition in psychometrics
  - 1970: PARAFAC (*parallel factor*) decomposition by Harshman in phonetics, CANDECOMP (*canonical decomposition*) by Carroll & Chang in psychometrics, a.k.a. CP (*CANDECOMP/PARAFAC*) by Kiers (2000)
- PARFAFAC/CP invented by Hitchcock in 1927: seminal idea of polyadic form of a tensor (sum of rank-one components) → *canonical polyadic decomposition* (CPD)

Some history facts (cont’d)

- From the 90s: Tensor decompositions were used in:
  - Chemistry, especially in chemometrics (Bro’s Ph.D. thesis, 1998)
  - Signal processing (blind source separation (BSS) using cumulant tensors (J.F. Cardoso, P. Comon, 1990, L. De Lathauwer, 1997)
- Since 2000: Tensor decompositions introduced in wireless communication problems (N. Sidiropoulos et al., 2000), and image analysis (Vasilescu & Terzopoulos, 2002)
- Last two decades: Tensor-based signal processing (wireless communications, antena array processing, image, speech processing, big data processing/analysis
- More recently: Numerous applications in machine learning/artificial intelligence (ML/AI)

Motivation

- Separation of data sets into components/factors to extract the multimodal structure of data and useful information from noisy measurements

- Dimensionality reduction of multidimensional data
  ⇒ Approximate low-rank tensor decompositions/models
  ⇒ Tensor train decompositions (massive datasets)

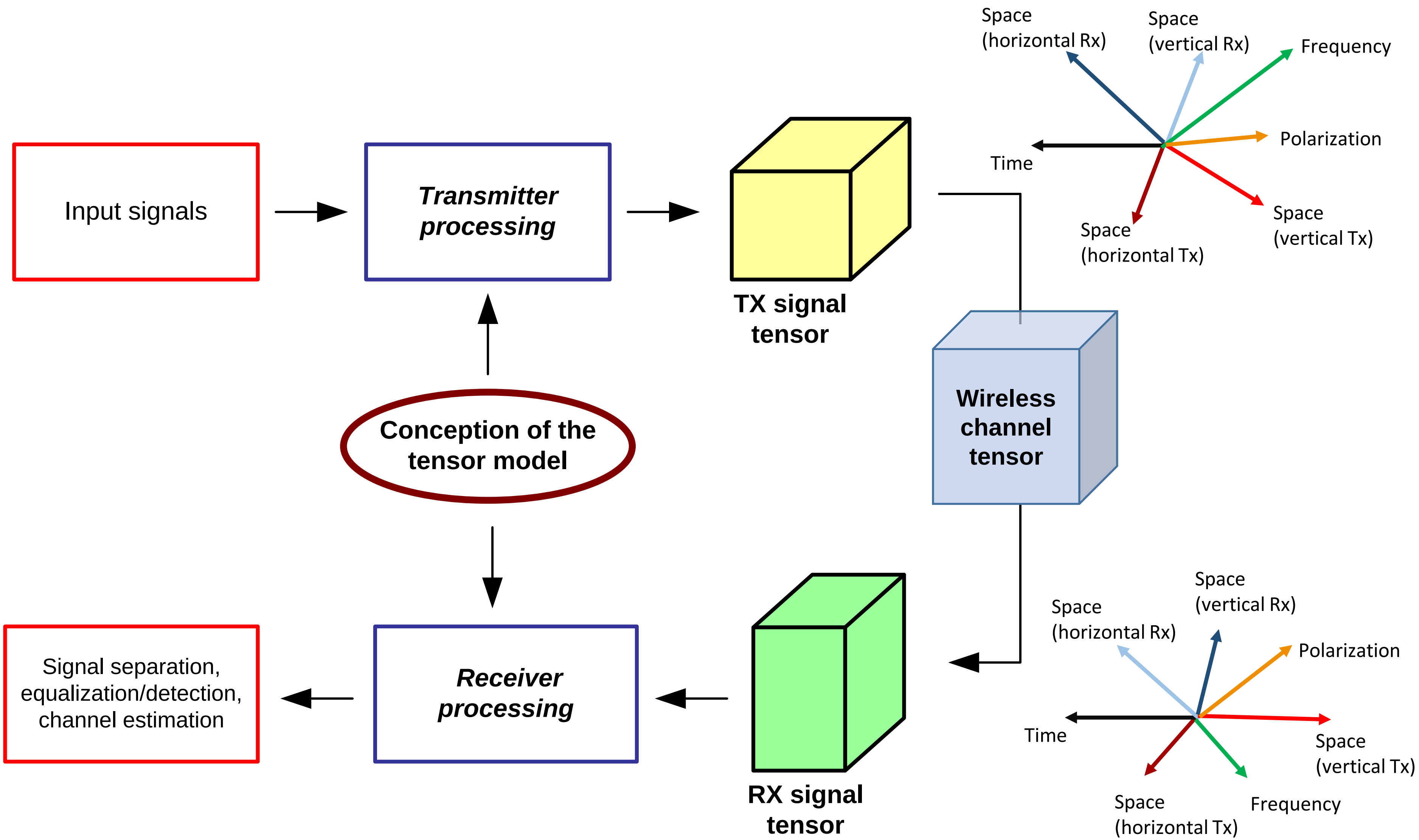
- Completion of data tensors in presence of missing data
  ⇒ New optimization problems and tensor-based algorithms

- Dynamic/streaming tensor analysis
  ⇒ Tensor factorization algorithms for high-order/large-scale tensors in distributed setup (parallel computing, tensor tracking, etc.)

Motivation (signal processing & communications)

- Exploit the multidimensional nature of the wireless channel and its multiple forms of diversity
- Blind/semi-blind channel estimation & symbol detection under more relaxed conditions (compared to matrix-based SP)
- Complexity reduction of large-scale filter optimizations (e.g. massive antenna arrays, equalizers, nonlinear filtering, neural network structures)
- Noise-relisient & robust multilinear modulation (low-rank tensor construction of the transmitted signals)

Tensor perspective to wireless communications



Motivation (signal processing & communications)

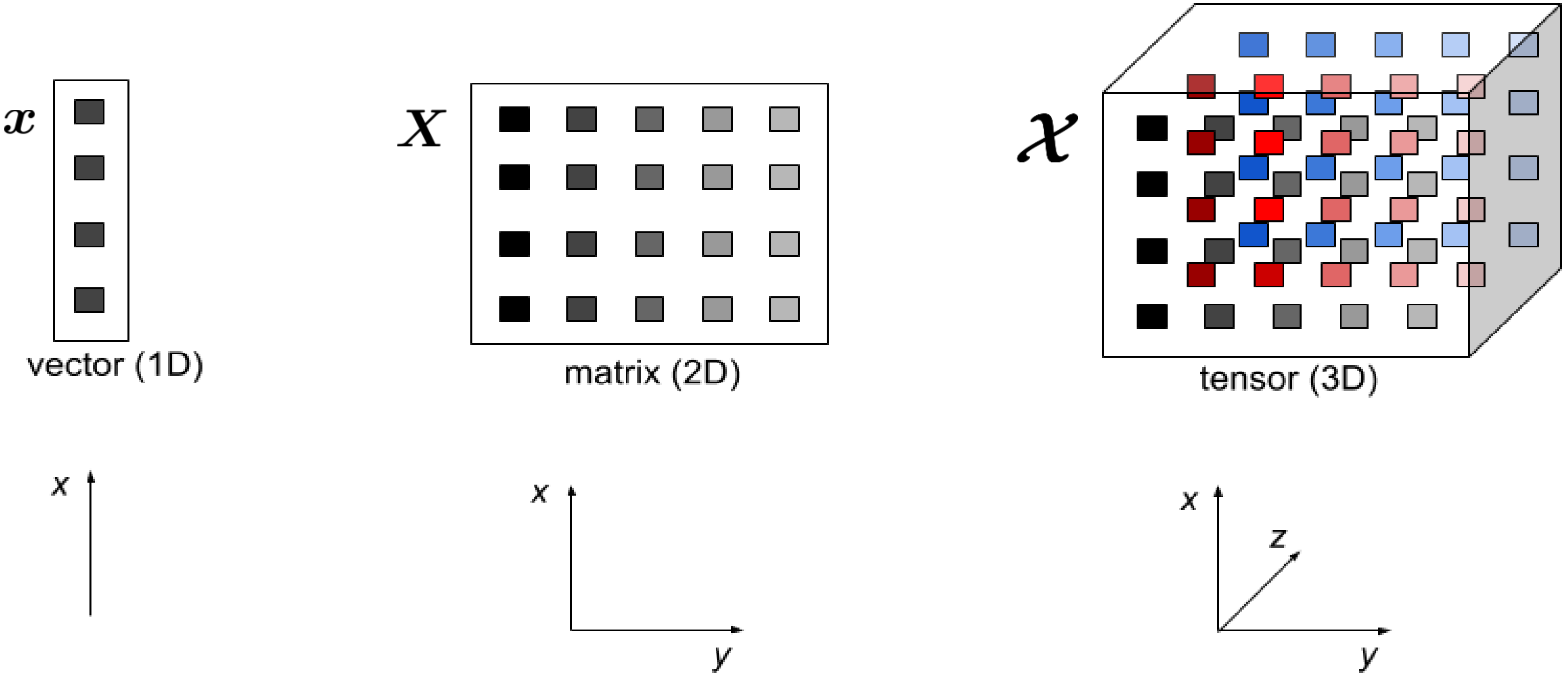
- Exploit the multidimensional nature of the wireless channel and its multiple forms of diversity
- Blind/semi-blind channel estimation & symbol detection under more relaxed conditions (compared to matrix-based SP)
- Complexity reduction of large-scale filter optimizations (e.g. massive antenna arrays, beamforming, equalizers, neural network structures)
- Noise-relisient & robust multilinear modulation (low-rank tensor construction of the transmitted signals)

PART 1

Tensor decompositions

What is a Tensor?

- An intuitive definition…



What is a tensor?

- A “nicer” mathematical definition 🙂

$$\boldsymbol{\mathcal{X}} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} x_{i,j,k} \underbrace{(\boldsymbol{e}_i^{(I)} \circ \boldsymbol{e}_j^{(J)} \circ \boldsymbol{e}_k^{(K)})}_{(i,j,k)\text{-th coordinate}}$$

$$\boldsymbol{e}_i^{(I)} = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i\text{-th position}$$

$\circ$: outer product

- Tensor as a multi-linear mapping

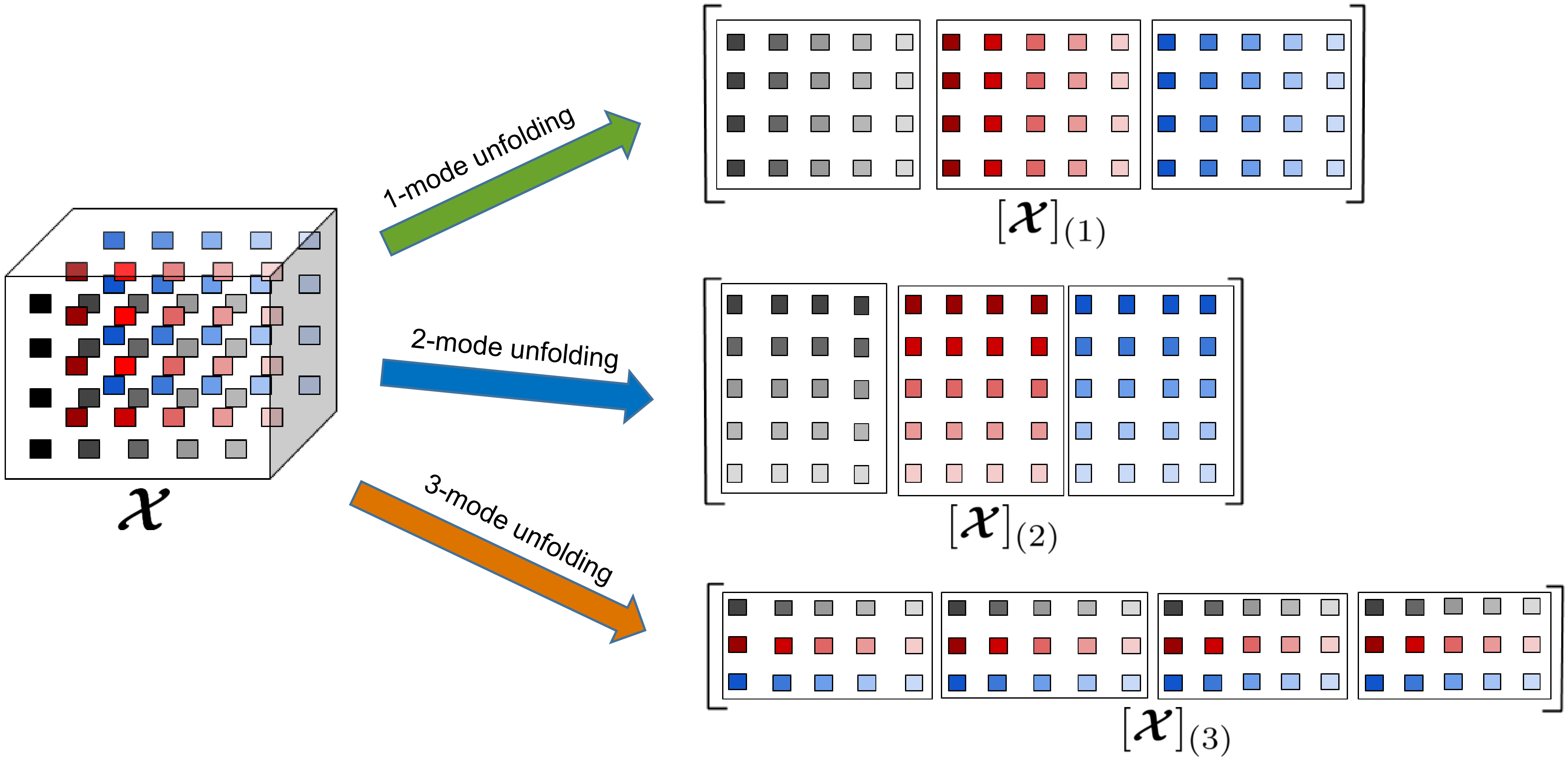
$$T(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}) = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} x_{i,j,k} (\boldsymbol{u}_i \circ \boldsymbol{v}_j \circ \boldsymbol{w}_k)$$

$$\boldsymbol{U} = [\boldsymbol{u}_i]$$

$$\boldsymbol{V} = [\boldsymbol{v}_j]$$

$$\boldsymbol{W} = [\boldsymbol{w}_k]$$

Unfolding a tensor into matrices



An useful operator: The *n*-mode product

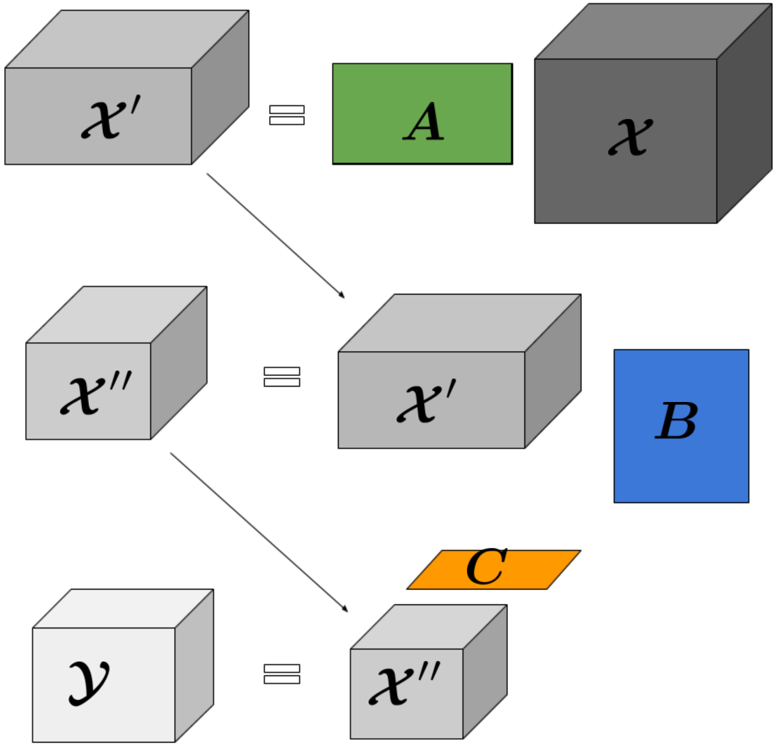
- Defines a product between a tensor and a matrix (or vector)

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{A} \quad \leftrightarrow \quad [\boldsymbol{\mathcal{Y}}]_{(n)} = \boldsymbol{A}[\boldsymbol{\mathcal{X}}]_{(n)}, \qquad \forall n$$

[De Lathauwer et al. '2000]
[Kolda & Bader, 2009]

- Multiple *n*-mode products

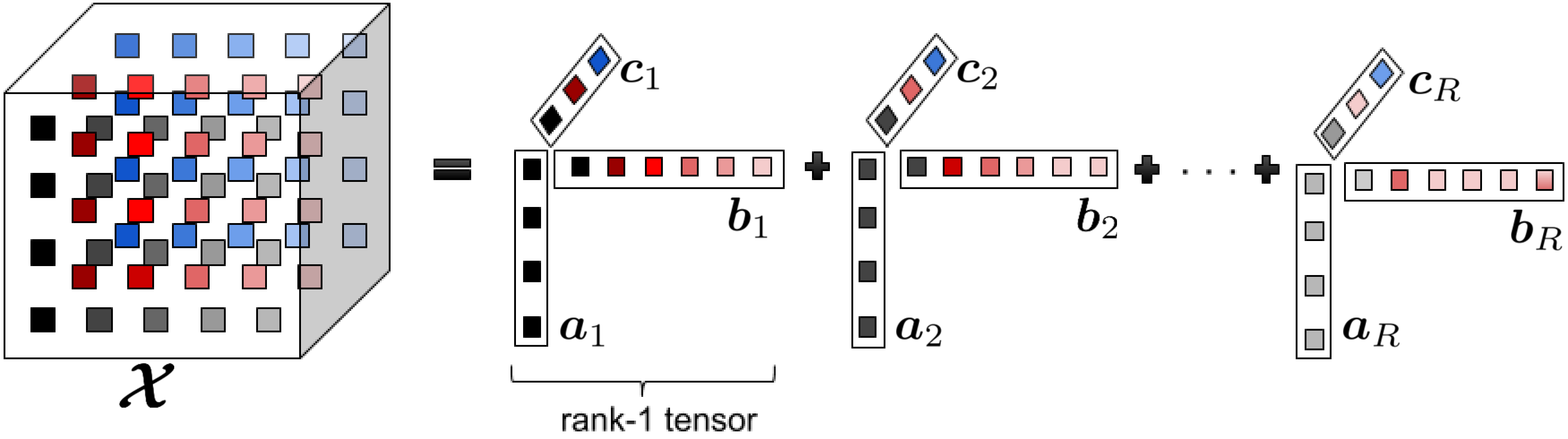
$$\boldsymbol{\mathcal{Y}} = (\underbrace{(\underbrace{(\underbrace{\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A})}_{\boldsymbol{\mathcal{X}}'} \times_2 \boldsymbol{B})}_{\boldsymbol{\mathcal{X}}''} \times_3 \boldsymbol{C})}_{\boldsymbol{\mathcal{Y}}}$$



Concept of "multi-linear compression"

The “canonical” tensor decomposition

- Decomposition in a mimimal sum of rank-1 components



Also known as:

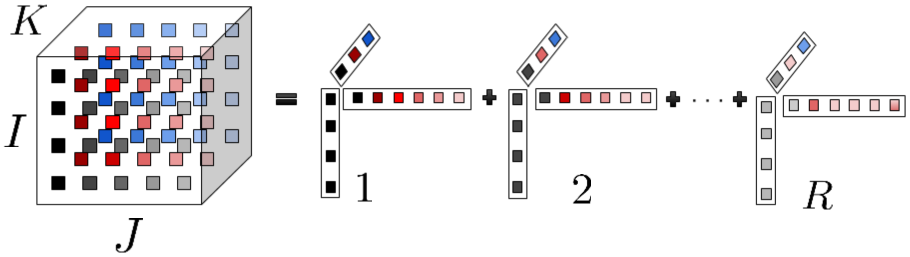
- Canonical polyadic decomposition (CPD) [Hithcock’1927]
- Parallel Factor decomposition (PARAFAC) [Harshman’1970] [Carroll & Chang’1970]

Tensor rank R → minimum # of rank-1 tensors yielding $\boldsymbol{\mathcal{X}}$ in a combination

Canonical polyadic decomposition (CPD)

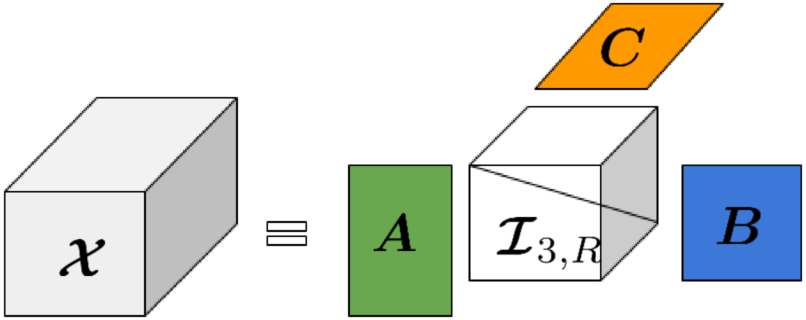
- Outer-product notation

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$$



- *n*-mode product notation

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,R} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$$



- "Vectorized" form

$$\boldsymbol{x} = (\boldsymbol{A} \diamond \boldsymbol{B} \diamond \boldsymbol{C})\boldsymbol{1}_R$$

$\diamond$: Khatri-Rao product

$$\boldsymbol{A} = [\boldsymbol{a}_r]$$

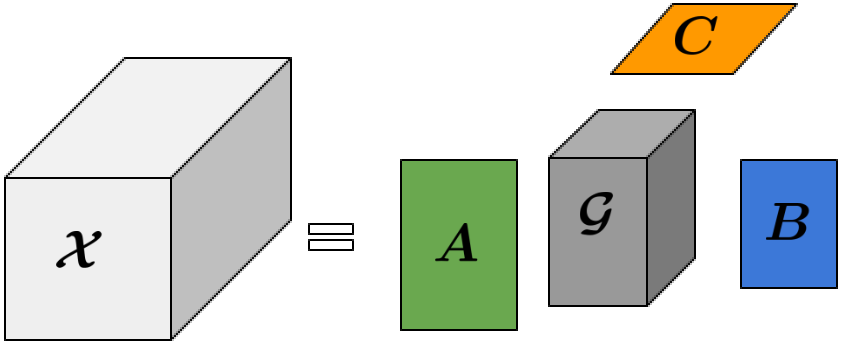
$$\boldsymbol{B} = [\boldsymbol{b}_r]$$

$$\boldsymbol{C} = [\boldsymbol{c}_r]$$

Tucker decomposition

Full multi-linear map $\mathcal{X} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{s=1}^{S} g_{p,q,s}(\boldsymbol{a}_p \circ \boldsymbol{b}_p \circ \boldsymbol{q}_s)$

[Tucker'1966]



$$\boldsymbol{A} = [\boldsymbol{a}_p]$$
$$\boldsymbol{B} = [\boldsymbol{b}_q]$$
$$\boldsymbol{C} = [\boldsymbol{c}_s]$$

- *n*-mode product notation

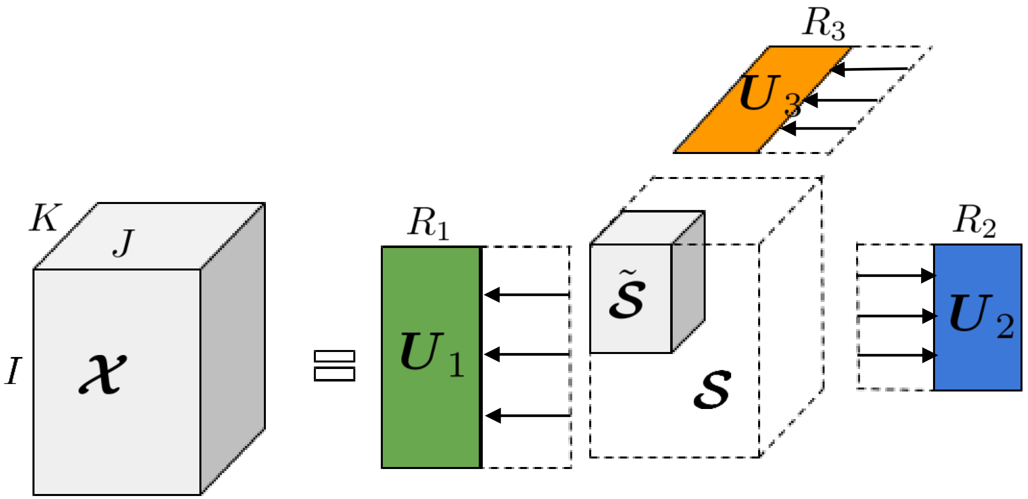
$$\mathcal{X} = \mathcal{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$$

- “Vectorized” form

$$\boldsymbol{x} = (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C})\boldsymbol{g}$$

High-order SVD (HOSVD)

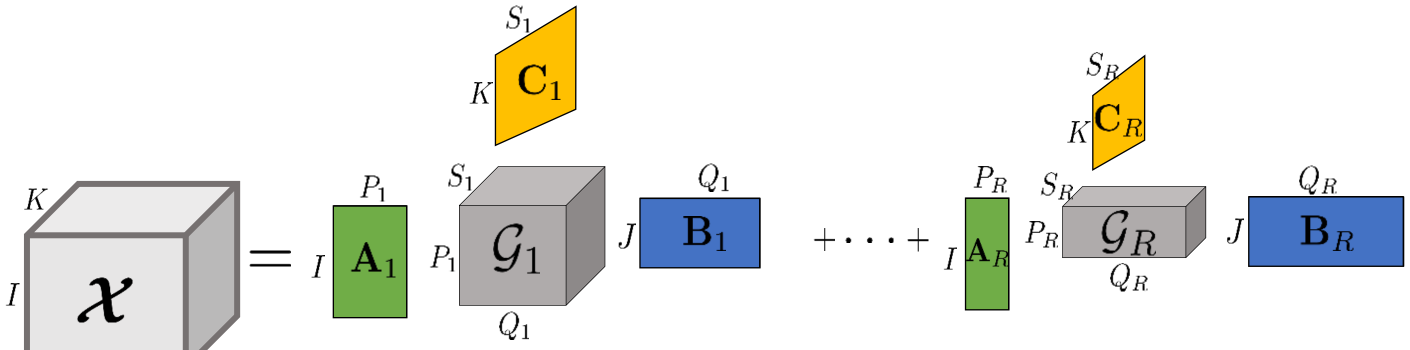
- Generalization of matrix SVD to tensors [De Lathauwer et al. '2000]



$$\tilde{\mathcal{X}} = \tilde{\mathcal{S}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3$$

Block term decomposition (BTD)

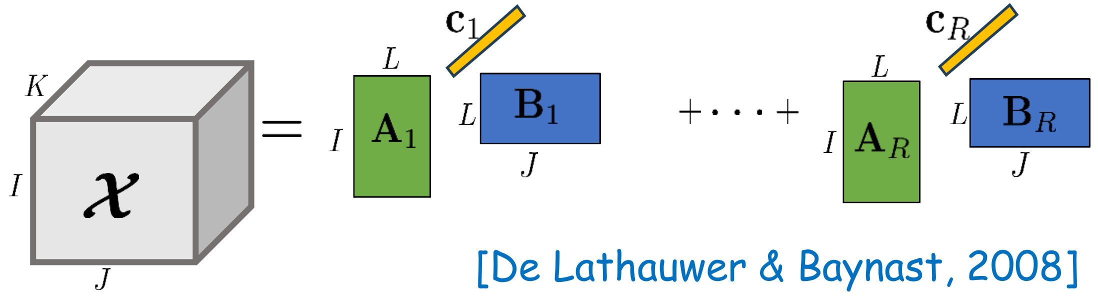
- Decomposition of a tensor into a sum of tensor "blocks" having lower multilinear ranks



[De Lathauwer, 2008]

$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{G}_r \times_1 \boldsymbol{A}_r \times_2 \boldsymbol{B}_r \times_3 \boldsymbol{C}_r$$

Special case: decomposition into rank-(*L*,*L*,1) blocks



$$\mathcal{X} = \sum_{r=1}^{R} (\boldsymbol{A}_r \boldsymbol{B}_r) \circ \boldsymbol{c}_r$$

[De Lathauwer & Baynast, 2008]

N-th order Tucker & Tucker-(N1,N)

- General expression:

$$x_{i_1,\dots,i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1,\dots,r_N} \prod_{n=1}^{N} a_{i_n,r_n}^{(n)} \rightarrow \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_{n=1}^{N} \boldsymbol{A}^{(n)}$$

- Tucker-(N1,N):

$$x_{i_1,\dots,i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{N_1}=1}^{R_{N_1}} g_{r_1,\dots,r_{N_1},i_{N_1+1},\dots,i_N} \prod_{n=1}^{N_1} a_{i_n,r_n}^{(n)}$$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \dots \times_{N_1} \boldsymbol{A}^{(N_1)} \times_{N_1+1} \boldsymbol{I}_{N_1+1} \times_{N_1+2} \dots \times_N \boldsymbol{I}_N$$

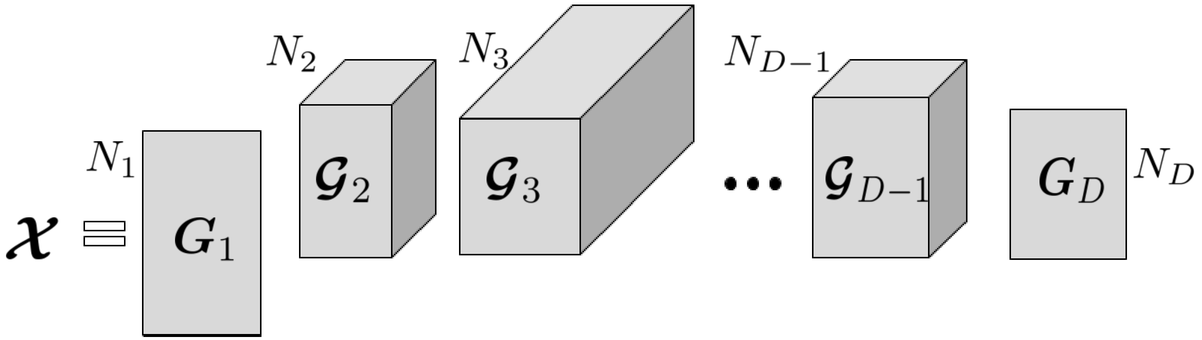
$$= \boldsymbol{\mathcal{G}} \times_{n=1}^{N_1} \boldsymbol{A}^{(n)}$$

Tensor Train (TT) decomposition

- D-dimensional tensor as a "train" of smaller 3D tensors

$\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$

[Oseledets, 2011]

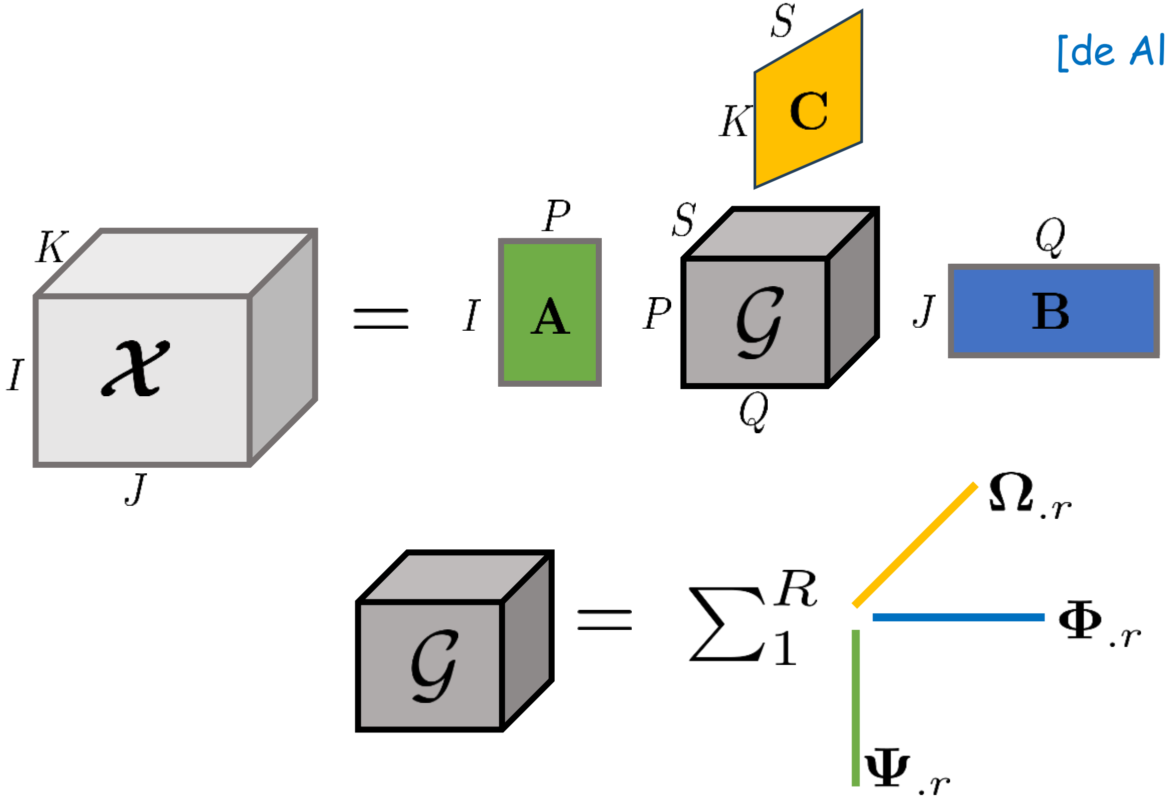


$$\boldsymbol{\mathcal{X}} = \boldsymbol{G}_1 \times_2^1 \boldsymbol{\mathcal{G}}_2 \times_3^1 \boldsymbol{\mathcal{G}}_3 \times_4^1 \cdots \times_{D-1}^1 \boldsymbol{\mathcal{G}}_{D-1} \times_D^1 \boldsymbol{G}_D$$

Introduced to tackle the curse of dimensionality
(case of "big data" tensors)

CONstrained FACtor decomposition (CONFAC)

[de Almeida *et al*, 2008]



CONFAC decomposition → Tucker-3 decomposition with "canonical" core tensor (PARAFAC-core)

UNIVERSIDADE FEDERAL DO CEARÁ

CONFAC decomposition (cont'd)

- Scalar writing:

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{s=1}^{S} a_{i,p} b_{j,q} c_{k,s} g_{p,q,s}(\mathbf{\Psi}, \mathbf{\Phi}, \mathbf{\Omega})$$

where $g_{p,q,s} = \sum_{r=1}^{R} \psi_{p,r}\phi_{q,r}\omega_{s,r}$ and $R = \max(P, Q, S)$

Columns of the constraint matrices $\mathbf{\Psi}$, $\mathbf{\Phi}$, and $\mathbf{\Omega}$ are canonical basis vectors (1's and 0's)

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$$

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{I}}_R \times_1 \mathbf{\Psi} \times_2 \mathbf{\Phi} \times_3 \mathbf{\Omega}$$

Tucker-3 with sparse PARAFAC core

CONFAC decomposition (cont'd)

- Interpretation as a rank-*R* "constrained" CPD

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{s=1}^{S}\Big(\sum_{r=1}^{R}\psi_{p,r}\phi_{q,r}\omega_{s,r}\Big)a_{i,p}b_{j,q}c_{k,s}$$

$$= \sum_{r=1}^{R}\Big(\sum_{p=1}^{P}a_{i,p}\psi_{p,r}\Big)\Big(\sum_{q=1}^{Q}b_{j,q}\phi_{q,r}\Big)\Big(\sum_{s=1}^{S}c_{k,s}\omega_{s,r}\Big)$$

⇕

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}} \times_1 (\boldsymbol{A}\boldsymbol{\Psi}) \times_2 (\boldsymbol{B}\boldsymbol{\Phi}) \times_3 (\boldsymbol{C}\boldsymbol{\Omega})$$

PARAFAC:

$$R_1 = R_2 = R_3 = F$$

$$\boldsymbol{\Psi} = \boldsymbol{\Phi} = \boldsymbol{\Omega} = \mathbf{I}_Q$$

$$\mathcal{G}(\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega}) = \mathcal{I}_Q$$

CONFAC decomposition (cont'd)

- Class of PARALIND models [Bro'2009]
- Enjoy partial uniqueness at different levels

  [Stegeman & de Almeida '2009] [Miron & Brie, 2015]
  [Guo *et al*, 2012]

- Essential uniqueness result [Stegeman & de Almeida, 2009]

Assumptions: $\mathbf{A}, \mathbf{B}, \mathbf{C}$ full column rank; $(\boldsymbol{\Phi} \diamond \boldsymbol{\Omega})\boldsymbol{\Psi}^T$ full column rank

$$N^* = \max_r \Big(\mathrm{rank}(\boldsymbol{\Phi}\mathrm{diag}(\boldsymbol{\psi}_r^T)\boldsymbol{\Phi}^T)\Big)$$

If $\mathrm{rank}(\boldsymbol{\Phi}\mathrm{diag}(\boldsymbol{\Psi}^T\mathbf{d})\boldsymbol{\Phi}^T)\Big) \leq N^*$ implies $\omega(\mathbf{d}) \leq 1$ ⇨ $\mathbf{A}$ is unique

PARALIND/CONFAC-(N1,N) decompositions

- Variant of PARALIND/CONFAC with only N1 constrained factor matrices [Favier & de Almeida, 2014]

$$x_{i_1,\ldots,i_{N_1+1},\ldots,i_N} = \sum_{f=1}^{F} \sum_{r_1=1}^{R_1} \cdots \sum_{r_{N_1}=1}^{R_{N_1}} \prod_{n=1}^{N_1} a_{i_n,r_n}^{(n)} \phi_{r_n,f}^{(n)} \prod_{n=N_1+1}^{N} a_{i_n,f}^{(n)}$$

(Tucker-(*N1*,*N*) with a “PARAFAC-like” core)

⇕

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{N,R} \times_{n=1}^{N_1} \left(\boldsymbol{A}^{(n)} \boldsymbol{\Phi}^{(n)}\right) \times_{n=N_1+1}^{N} \boldsymbol{A}^{(n)}$$

Constraints only affect the first N1 modes while the other are “free” modes

Block PARALIND/CONFAC

- Block-partitioned version of PARALIND/CONFAC

$$\boldsymbol{\mathcal{X}} = \sum_{p=1}^{P} \boldsymbol{\mathcal{X}}_p \quad \text{with} \quad \boldsymbol{\mathcal{X}}_p = \boldsymbol{\mathcal{G}}_p \times_{n=1}^{N} \boldsymbol{A}_p^{(n)}$$

$$\boldsymbol{\mathcal{G}}_p = \boldsymbol{\mathcal{I}}_{N,R_p} \times_{n=1}^{N} \boldsymbol{\Phi}_p^{(n)}$$

Special case: Block CONFAC-(1,3)

Fixed constraint in only one mode (*N1*=1, *N*=3)

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{N,R} \times_1 (\boldsymbol{A}\boldsymbol{\Phi}) \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} = \sum_{r=1}^{R} \boldsymbol{a}_r \circ (\boldsymbol{B}_r \boldsymbol{C}_r)$$

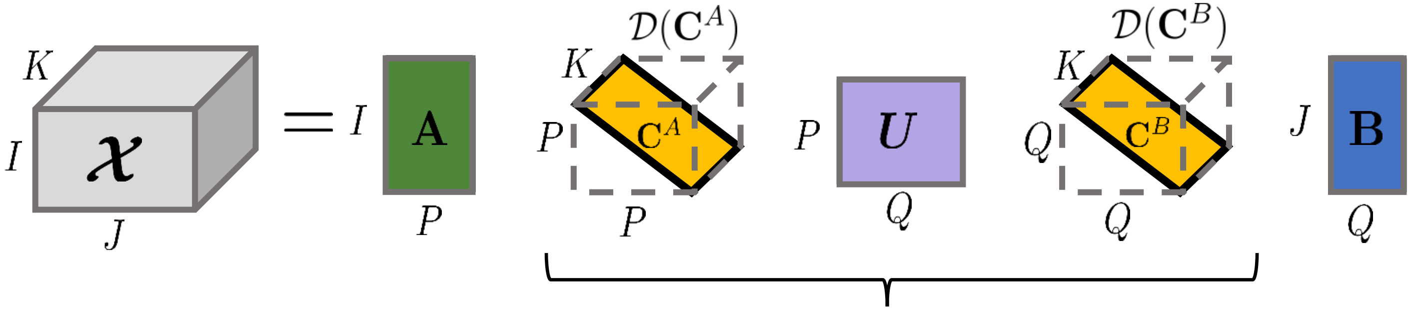
$$\boldsymbol{\Psi} \doteq \mathrm{diag}(\mathbf{1}_{L_1}^{\mathrm{T}}, \ldots, \mathbf{1}_{L_P}^{\mathrm{T}})$$

$$\boldsymbol{A} \doteq [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_P]$$
$$\boldsymbol{B} \doteq [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_P]$$
$$\boldsymbol{C} \doteq [\boldsymbol{C}_1, \ldots, \boldsymbol{C}_P]$$

Block CONFAC-(1,3) → rank-(Lp,Lp,1) BTD

PARATUCK-type decompositions

- The PARATUCK-2 decomposition [Harshman & Lundy, 1996]



Tucker-(2,3) with a structured core tensor

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q} \underbrace{\left(u_{p,q}c^{A}_{p,k}c^{B}_{q,k}\right)}_{w_{p,q,k}} a_{i,p}b_{j,q}$$

$$= \sum_{p=1}^{P}\sum_{q=1}^{Q} u_{p,q}\left(a_{i,p}c^{A}_{p,k}\right)\left(b_{j,q}c^{B}_{q,k}\right)$$

Interpretation of C^A and C^B: *interaction* or *allocation* matrices

PARATUCK-type decompositions (cont'd)

- PARATUCK-2 as a hybrid of PARAFAC and Tucker-2

$$x_{i,j,k} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \underbrace{(u_{p,q} c^{A}_{p,k} c^{B}_{q,k})}_{w_{p,q,k}} a_{i,p} b_{j,q} \longrightarrow \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{W}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B}$$

Where is the PARAFAC structure?

[Favier & de Almeida, 2014]

Defining

$$\begin{cases} f_{p,q,k} = c^{A}_{p,k} c^{B}_{q,k} = \sum_{j=1}^{K} c^{A}_{p,j} c^{B}_{q,j} \delta_{k,j} \\ \Updownarrow \\ \boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{I}}_{3,K} \times_1 \boldsymbol{C}^{A} \times_2 \boldsymbol{C}^{B} \times_3 \boldsymbol{I}_K \end{cases}$$

$$\boldsymbol{\mathcal{X}} = \underbrace{\boldsymbol{\mathcal{W}}}_{\boldsymbol{U} \underset{\{p,q\}}{\odot} \boldsymbol{\mathcal{F}}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \longrightarrow$$

PARATUCK-2: Tucker-2 with (hidden) PARAFAC-core tensor

PARATUCK-2 decomposition

- PARATUCK-2 as a “structured” Tucker-3 [Sokal *et al*, 2020]

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q} u_{p,q}(a_{i,p}c^{A}_{p,k})(b_{j,q}c^{B}_{q,k})$$

tensorize

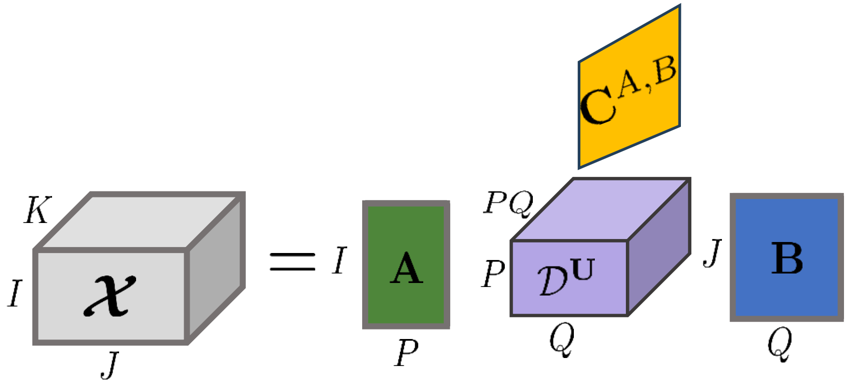
merge

$$\begin{cases} c^{AB}_{k,r} \doteq c^{A}_{p,k}c^{B}_{q,k} \\ r \doteq (p-1)Q+q \end{cases}$$

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{PQ} d^{U}_{p,q,r}\, a_{i,p}\, b_{j,q}\, c^{AB}_{k,r}$$

$$\Updownarrow$$

$$\mathcal{X} = \mathcal{D}^{\mathbf{U}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}^{A,B}$$



Sparse core tensor

$$[\mathcal{D}^{\mathbf{U}}]_{(3)} \doteq \mathbf{U}^{\mathrm{T}}$$

Khatri-Rao-structured factor matrix

$$\mathbf{C}^{A,B} \doteq [(\mathbf{C}^{A})^{\mathrm{T}} \diamond (\mathbf{C}^{B})^{\mathrm{T}}]^{\mathrm{T}}$$

PARATUCK-(2,4) and PARATUCK-(*N1*,*N*)

- PARATUCK-(2,4) decomposition

[da Costa et al, 2011]
[de Araújo & de Almeida, 2022]

$$x_{i,j,k,l} = \sum_{p=1}^{P}\sum_{q=1}^{Q} \underbrace{(u_{p,q,l}c^{A}_{p,k}c^{B}_{q,k})}_{w_{p,q,l,k}} a_{i,p}b_{j,q}$$

$$= \sum_{p=1}^{P}\sum_{q=1}^{Q} u_{p,q,l}(a_{i,p}c^{A}_{p,k})(b_{j,q}c^{B}_{q,k})$$

$\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I\times J\times K\times L}$

Tucker-(2,4) with structured core tensor

- PARATUCK-(*N1*,*N*) decomposition

[Favier & de Almeida, 2014]

$$x_{i_1,\ldots,i_{N_1+1},\ldots,i_N} = \sum_{r_1=1}^{R_1}\cdots\sum_{r_{N_1}=1}^{R_{N_1}} u_{r_1,\ldots,r_{N_1},i_{N_1+2},\ldots,i_N}\prod_{n_1=1}^{N_1} a^{(n)}_{i_n,r_n}c^{(n)}_{r_n,i_{N_1+1}}$$

$a^{(n)}_{i_n,r_n}, c^{(n)}_{r_n,i_{N_1+1}}$ are entries of the factor matrix $\boldsymbol{A}^{(n)} \in \mathbb{C}^{I_n\times R_n}$ and the interaction matrix $\boldsymbol{C}^{(n)} \in \mathbb{C}^{R_n\times I_{N_1+1}}, \forall n = 1,\ldots,N_1$

Links with constrained PARAFAC decompositions

- PARATUCK-2 as constrained PARAFAC-3 [Favier & de Almeida, 2014]

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q} a_{i,p} b_{j,q} (u_{p,q} c^{A}_{p,k} c^{B}_{q,k})$$

Defining $\begin{cases} \boldsymbol{\Psi}^{A} \doteq \boldsymbol{I}_P \otimes \mathbf{1}_Q^{\mathrm{T}} \\ \boldsymbol{\Psi}^{B} \doteq \mathbf{1}_P^{\mathrm{T}} \otimes \boldsymbol{I}_Q \end{cases}$ → constraint matrices

Equivalent expression:

$$x_{i,j,k} = \sum_{r=1}^{PQ}\left(\sum_{p=1}^{P} a_{i,p}\psi^{A}_{p,r}\right)\left(\sum_{q=1}^{Q} b_{j,q}\psi^{B}_{q,r}\right)(u_{p,q} c^{A}_{p,k} c^{B}_{q,k})$$

$$\Updownarrow$$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,PQ} \times_1 (\boldsymbol{A}\boldsymbol{\Psi}^{A}) \times_2 (\boldsymbol{B}\boldsymbol{\Psi}^{B}) \times_3 \boldsymbol{F}^{AB}$$

Constrained PARAFAC-3 decomp.
(special CONFAC-(2,3) case)

with $\boldsymbol{F}^{AB} = [\boldsymbol{C}^{A} \diamond \boldsymbol{C}^{B}]^{\mathrm{T}} \mathrm{diag}(\mathrm{vec}(\boldsymbol{U}))$

Links with constrained PARAFAC decompositions

- PARATUCK-(2,4) as constrained PARAFAC-4 [Favier & de Almeida, 2014]

$$x_{i,j,k} = \sum_{p=1}^{P}\sum_{q=1}^{Q} a_{i,p} b_{j,q} (u_{p,q,l} c^{A}_{p,k} c^{B}_{q,k})$$

Defining $\begin{cases} \boldsymbol{\Psi}^{A} \doteq \boldsymbol{I}_P \otimes \mathbf{1}_Q^{\mathrm{T}} \\ \boldsymbol{\Psi}^{B} \doteq \mathbf{1}_P^{\mathrm{T}} \otimes \boldsymbol{I}_Q \end{cases}$ and $\boldsymbol{D} \doteq [\boldsymbol{\mathcal{U}}]_{(3)} (L \times PQ)$

Equivalent expression:

$$x_{i,j,k} = \sum_{r=1}^{PQ} \left(\sum_{p=1}^{P} a_{i,p} \psi^{A}_{p,r} \right) \left(\sum_{q=1}^{Q} b_{j,q} \psi^{B}_{q,r} \right) (c^{A}_{p,k} c^{B}_{q,k}) d_{l,r}$$

$$\Updownarrow$$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,PQ} \times_1 (\boldsymbol{A}\boldsymbol{\Psi}^{A}) \times_2 (\boldsymbol{B}\boldsymbol{\Psi}^{B}) \times_3 \boldsymbol{F}^{AB} \times_4 \boldsymbol{D}$$

with $\boldsymbol{F}^{AB} = [\boldsymbol{C}^{A} \diamond \boldsymbol{C}^{B}]^{\mathrm{T}}$

Constrained PARAFAC-4 decomp.
(special CONFAC-(2,4) case)

Links with constrained PARAFAC decompositions

- PARATUCK-(*N*-2,*N*) as constrained PARAFAC-*N* [Favier & de Almeida, 2014]

$$x_{i_1,\ldots,i_{N_1+1},\ldots,i_N} = \sum_{r_1=1}^{R_1}\cdots\sum_{r_{N_1}=1}^{R_{N_1}} u_{r_1,\ldots,r_{N_1},i_{N_1+2},\ldots,i_N}\prod_{n_1=1}^{N_1} a^{(n)}_{i_n,r_n}\, c^{(n)}_{r_n,i_{N_1+1}}$$

Defining
$$\begin{cases} \boldsymbol{\Psi}^{(n)} \doteq \mathbf{1}^{\mathrm{T}}_{R_1} \otimes \cdots \otimes \mathbf{1}^{\mathrm{T}}_{R_{n-1}} \otimes \boldsymbol{I}_{R_n} \otimes \mathbf{1}^{\mathrm{T}}_{R_{n+1}} \otimes \cdots \otimes \mathbf{1}^{\mathrm{T}}_{R_N} \\ \boldsymbol{D} \doteq [\boldsymbol{\mathcal{U}}]_{(N)} (I_N \times R) \\ \boldsymbol{F} = [\diamond_{n=1}^{N} \boldsymbol{C}^{(n)}]^{\mathrm{T}} \end{cases}$$

$$\begin{cases} r \doteq r_{N_1} + \sum_{n=1}^{N_1-1} (r_n - 1) \prod_{i=n+1}^{N_1} R_i \\ \text{and} \\ R \doteq \prod_{i=1}^{N_1} R_i \end{cases}$$

Equivalent expression:

$$x_{i_1,\ldots,i_N} = \sum_{r=1}^{R} \Big(\prod_{n=1}^{N-2} (a^{(n)}_{i_n,r} \psi^{(n)}_{r_n,r}) \Big) f_{i_{n-1},r} d_{i_n,r}$$

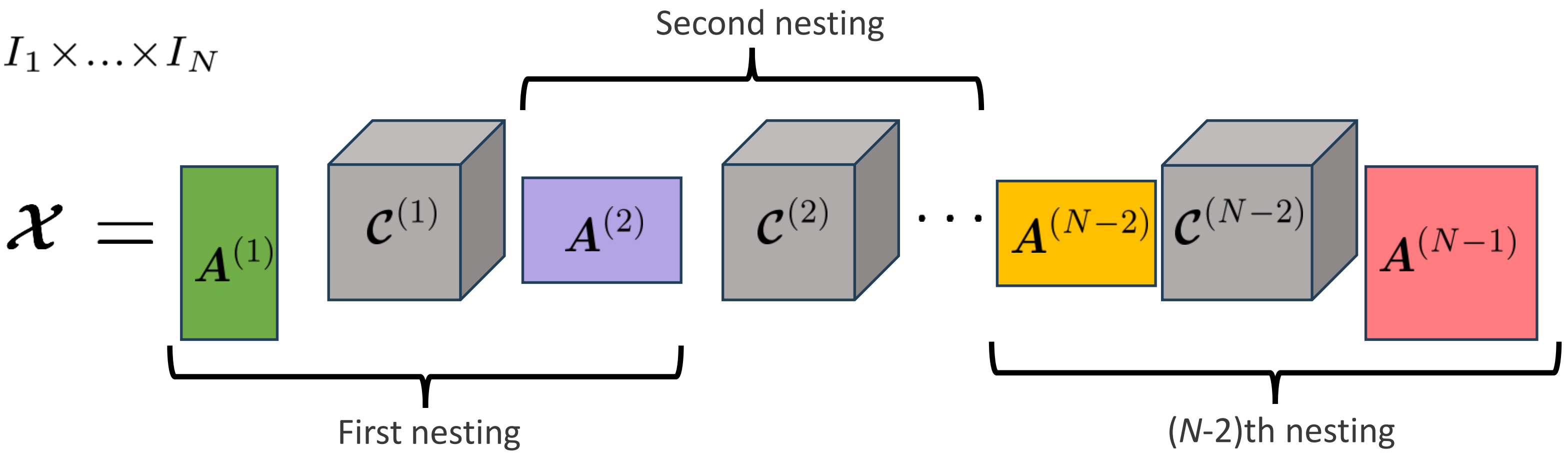
⇕

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,R} \times_{n=1}^{N-2} (\boldsymbol{A}^{(n)} \boldsymbol{\Psi}^{(n)}) \times_{N-1} \boldsymbol{F} \times_N \boldsymbol{D}$$

Constrained PARAFAC-*N* decomp.
(special CONFAC-(*N*-2,*N*) case)

Nested Tucker decomposition (NTD)

$\mathcal{X} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$

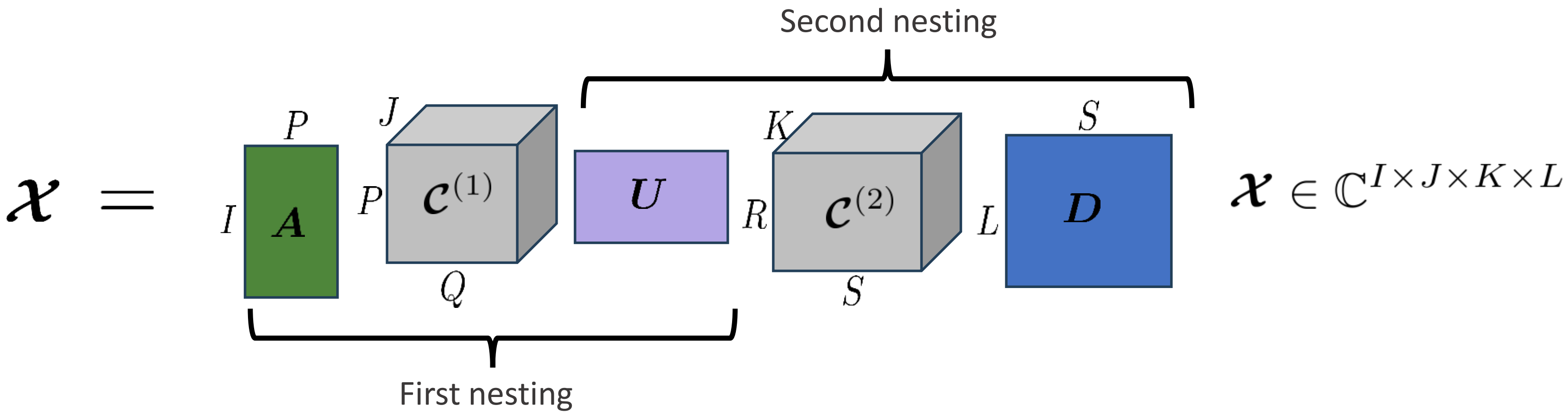


Each third-order tensor $\mathcal{C}^{(n)} \in \mathbb{C}^{R_{2n-1} \times I_{n+1} \times R_{2n}}$, $n \in [1, N-2]$ can be considered as a core tensor of a Tucker-$(2,3)$ term having $(\boldsymbol{A}^{(n)}, \boldsymbol{I}_{I_{n+1}}, \boldsymbol{A}^{(n+1)})$ as matrix factors, with:

$$\boldsymbol{A}^{(n+1)} \in \mathbb{C}^{R_{2n} \times R_{2n+1}},\ n \in [2, N-2], \quad \boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times R_1}, \boldsymbol{A}^{(N-1)} \in \mathbb{C}^{I_N \times R_{2N-4}}$$

Train of Tucker-(2,3) terms, where two successive terms share a common factor matrix

NTD-4 (case of 4th order tensor)

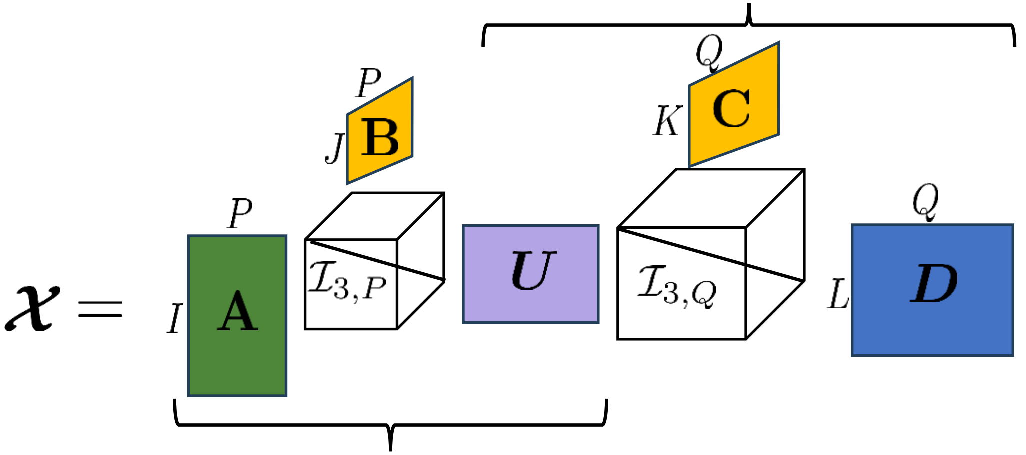


$$x_{i,j,k,l} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} \sum_{s=1}^{S} \underbrace{a_{i,p} c^{(1)}_{p,j,q} u_{q,r}}_{\text{Tucker-(2,3)}} c^{(2)}_{r,k,s} d_{l,s}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} \sum_{s=1}^{S} a_{i,p} c^{(1)}_{p,j,q} \underbrace{u_{q,r} c^{(2)}_{r,k,s} d_{l,s}}_{\text{Tucker-(2,3)}}$$

Nesting of two Tucker-(2,3) tensors that share a common factor matrix

Nested PARAFAC (case of 4th order tensor)



[de Almeida *et al*, 2013]

Special case of Nested Tucker (NTD-4) with the following correspondences:

$$(p, q, r, s) \Leftrightarrow (p, p, q, q)$$

$$(\boldsymbol{A}, \boldsymbol{\mathcal{C}}^{(1)}, \boldsymbol{U}, \boldsymbol{\mathcal{C}}^{(2)}, \boldsymbol{D}) \Leftrightarrow (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{U}, \boldsymbol{C}, \boldsymbol{D})$$

$$x_{i,j,k,l} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \underbrace{a_{i,p} b_{j,p} u_{p,q}}_{\text{PARAFAC}} c_{k,q} d_{l,q} = \sum_{p=1}^{P} \sum_{q=1}^{Q} a_{i,p} b_{j,q} \underbrace{u_{p,q} c_{k,q} d_{l,q}}_{\text{PARAFAC}}$$

Nesting of two PARAFAC tensors that share a common factor matrix

Nested PARAFAC (con’t)

Define the tensors $\boldsymbol{\mathcal{W}} \in \mathbb{C}^{K\times L\times P}$, $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{I\times J\times Q}$ such as

$$w_{k,l,p} = \sum_{q=1}^{Q} c_{k,q} d_{l,q} u_{p,q}$$

$$z_{i,j,q} = \sum_{p=1}^{P} a_{i,p} b_{j,p} u_{p,q}$$

or, equivalently in terms of mode-n products

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{I}}_{3,Q} \times_1 \boldsymbol{C} \times_2 \boldsymbol{D} \times_3 \boldsymbol{U}$$

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{I}}_{3,P} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{U}^{\mathrm{T}}$$

→ $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{Z}}$ satisfy two 3rd order PARAFAC models that share a common factor matrix

Nested PARAFAC (con't)

- Unfoldings of $\mathcal{W}$, $\mathcal{Z}$:

$$\mathcal{W} = \mathcal{I}_{3,Q} \times_1 \boldsymbol{C} \times_2 \boldsymbol{D} \times_3 \boldsymbol{U} \longrightarrow [\mathcal{W}]_{(3)} = \boldsymbol{U}(\boldsymbol{C} \diamond \boldsymbol{D})^{\mathrm{T}}$$

$$\mathcal{Z} = \mathcal{I}_{3,P} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{U}^{\mathrm{T}} \longrightarrow [\mathcal{Z}]_{(3)} = \boldsymbol{U}^{\mathrm{T}}(\boldsymbol{A} \diamond \boldsymbol{B})^{\mathrm{T}}$$

- Merging the last two modes, we get:

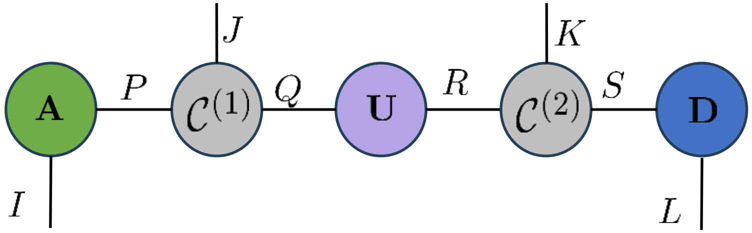
$$x^{(1)}_{i,j,t} = \sum_{p=1}^{P} a_{i,p} b_{j,p} w_{t,p} \Leftrightarrow \mathcal{X}^{(1)} = \mathcal{I}_{3,P} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \underbrace{(\boldsymbol{C} \diamond \boldsymbol{D})\boldsymbol{U}^{\mathrm{T}}}_{[\mathcal{W}]^{\mathrm{T}}_{(3)}}$$

- Merging the first two modes, we get:

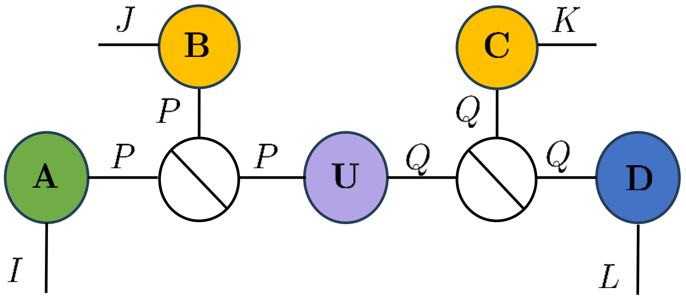
$$x^{(2)}_{m,k,l} = \sum_{q=1}^{Q} z_{m,q} c_{k,q} d_{l,q} \Leftrightarrow \mathcal{X}^{(2)} = \mathcal{I}_{3,Q} \times_1 \underbrace{(\boldsymbol{A} \diamond \boldsymbol{B})\boldsymbol{U}}_{[\mathcal{Z}]^{\mathrm{T}}_{(3)}} \times_2 \boldsymbol{C} \times_3 \boldsymbol{D}$$

→ $\mathcal{X}^{(1)}$, $\mathcal{X}^{(2)}$ satisfy two nested 3$^{\mathrm{rd}}$-order PARAFAC models

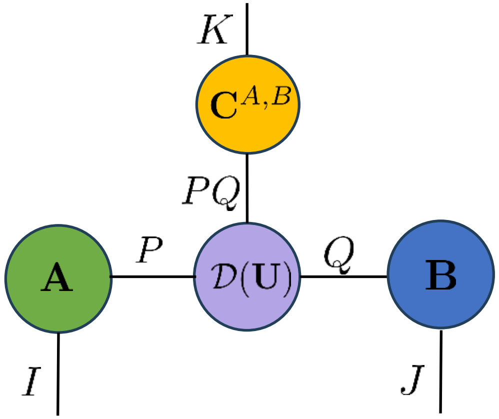
Comparisons using tensor network diagrams



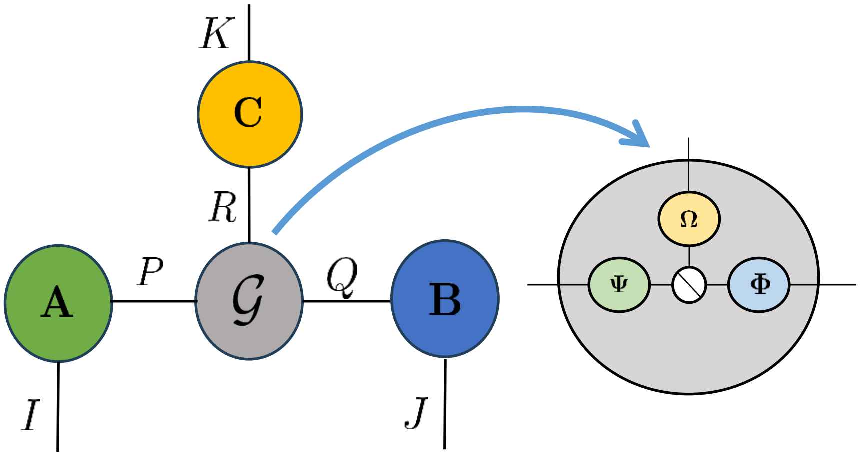
Nested Tucker-(2,4)



Nested PARAFAC-4



PARATUCK-(2,3)



CONFAC-3

PART 2

Some applications

Modeling/estimation of MIMO channels

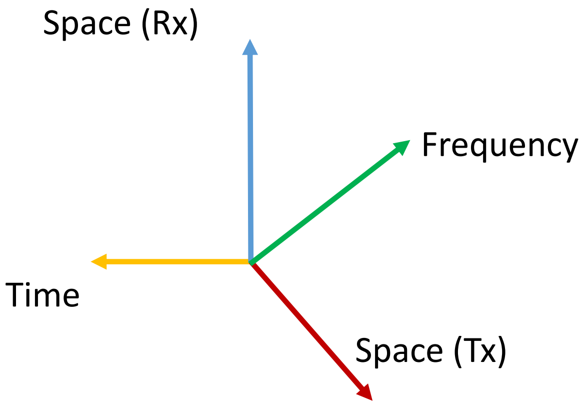
"Tensorizing" the MIMO channel model

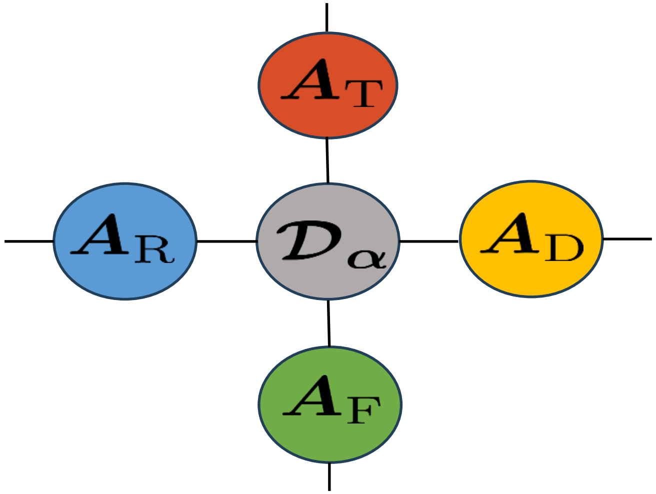
- Usual (matrix) notation

$$\boldsymbol{H}(t,f) = \sum_{\ell=1}^{N_{\mathrm{p}}} \alpha_\ell e^{j2\pi(\nu_\ell t - \tau_\ell f)} \boldsymbol{a}_{\mathrm{R}}(\theta_{R,\ell}, \phi_{R,\ell}) \boldsymbol{a}_{\mathrm{T}}^{*}(\theta_{T,\ell}, \phi_{T,\ell})$$

- Tensor notation (4D tensor, rank- N_{p})

$$\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{D}}_{\boldsymbol{\alpha}} \times_1 \boldsymbol{A}_{R}(\boldsymbol{\theta}_{\mathrm{R}}, \boldsymbol{\phi}_{\mathrm{R}}) \times_2 \boldsymbol{A}_{T}(\boldsymbol{\theta}_{\mathrm{T}}, \boldsymbol{\phi}_{\mathrm{T}}) \times_3 \boldsymbol{A}_{\mathrm{D}}(\boldsymbol{\nu}) \times_4 \boldsymbol{A}_{\mathrm{F}}(\boldsymbol{\tau})$$





"Tensorizing" the channel model (cont'd)

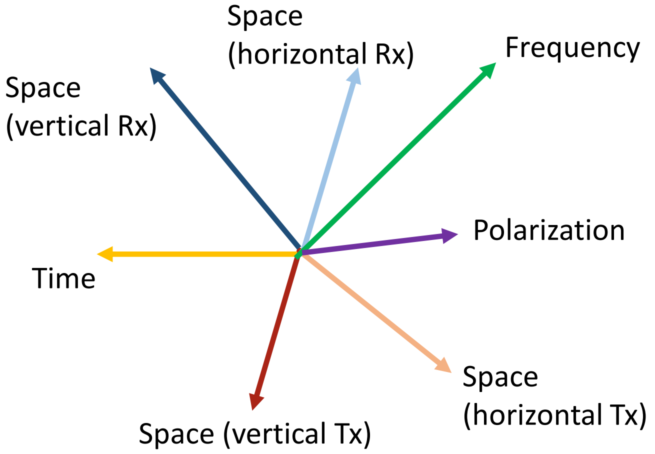
- Expanding the tensor (+ 2D antenna arrays, e.g. URA)

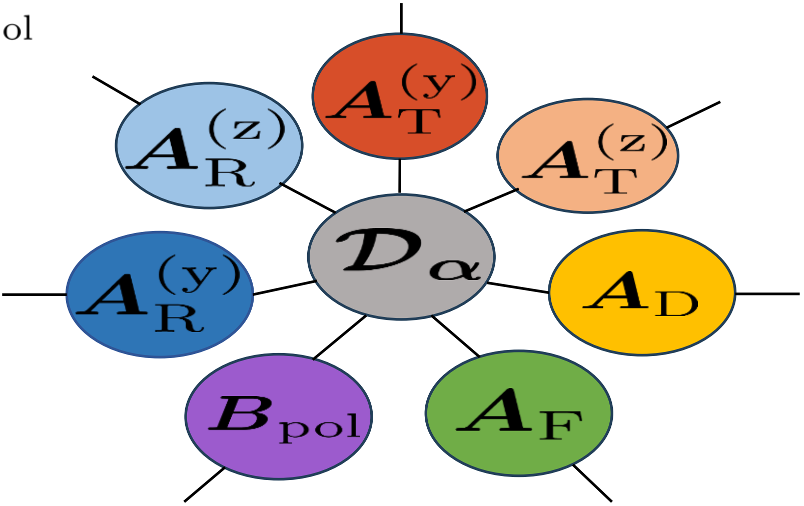
$$\boldsymbol{A}_{\mathrm{R}}(\boldsymbol{\theta}_{\mathrm{R}}, \boldsymbol{\phi}_{\mathrm{R}}) = \boldsymbol{A}_{\mathrm{R}}(\boldsymbol{\mu}_{\mathrm{R}}^{(\mathrm{y})}) \diamond \boldsymbol{A}_{\mathrm{R}}(\boldsymbol{\mu}_{\mathrm{R}}^{(\mathrm{z})}) \quad \boldsymbol{A}_{\mathrm{T}}(\boldsymbol{\theta}_{\mathrm{T}}, \boldsymbol{\phi}_{\mathrm{T}}) = \boldsymbol{A}_{\mathrm{T}}(\boldsymbol{\mu}_{\mathrm{T}}^{(\mathrm{y})}) \diamond \boldsymbol{A}_{\mathrm{T}}(\boldsymbol{\mu}_{\mathrm{T}}^{(\mathrm{z})})$$

$$\begin{aligned} \boldsymbol{\mathcal{H}} \quad = \quad & \boldsymbol{\mathcal{D}}_{\boldsymbol{\alpha}} \times_1 \boldsymbol{A}_{R}^{(\mathrm{y})}(\boldsymbol{\mu}_{\mathrm{R}}^{(\mathrm{y})}) \times_2 \boldsymbol{A}_{R}^{(\mathrm{z})}(\boldsymbol{\mu}_{\mathrm{R}}^{(\mathrm{z})}) \times_3 \boldsymbol{A}_{T}^{(\mathrm{y})}(\boldsymbol{\mu}_{\mathrm{T}}^{(\mathrm{y})}) \times_4 \boldsymbol{A}_{T}^{(\mathrm{z})}(\boldsymbol{\mu}_{\mathrm{T}}^{(\mathrm{z})}) \\ & \times_5 \boldsymbol{A}_{\mathrm{D}}(\boldsymbol{\nu}) \times_6 \boldsymbol{A}_{\mathrm{F}}(\boldsymbol{\tau}) \end{aligned}$$

- Expanding the tensor (+ polarization) → 7 dimensions

$$\begin{aligned} \boldsymbol{\mathcal{H}} \quad = \quad & \boldsymbol{\mathcal{D}}_{\boldsymbol{\alpha}} \times_1 \boldsymbol{A}_{R}(\boldsymbol{\mu}_{\mathrm{R}}^{(1)}) \times_2 \boldsymbol{A}_{R}(\boldsymbol{\mu}_{\mathrm{R}}^{(2)}) \times_3 \boldsymbol{A}_{T}(\boldsymbol{\mu}_{\mathrm{T}}^{(1)}) \times_4 \boldsymbol{A}_{T}(\boldsymbol{\mu}_{\mathrm{T}}^{(2)}) \\ & \times_5 \boldsymbol{A}_{\mathrm{D}}(\boldsymbol{\nu}) \times_6 \boldsymbol{A}_{\mathrm{F}}(\boldsymbol{\tau}) \times_7 \boldsymbol{B}_{\mathrm{pol}} \end{aligned}$$





Tensor Train Based Channel Estimation

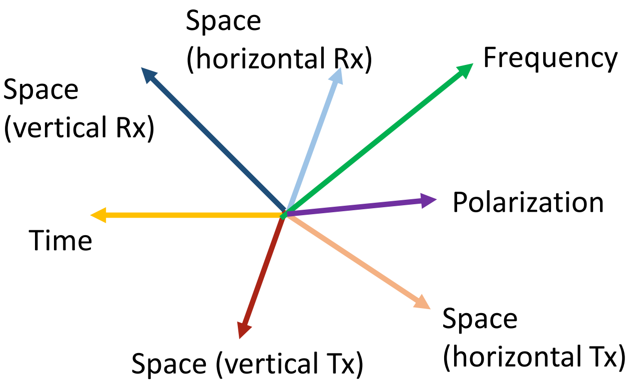
- MIMO channel (rectangular arrays, dual-polarized antennas)

$$\mathcal{H} = \mathcal{I}_{7,N_{\mathrm{p}}} \times_1 \boldsymbol{A}_{\mathrm{R}}^{(\mathrm{x})} \times_2 \boldsymbol{A}_{\mathrm{R}}^{(\mathrm{y})} \times_3 \boldsymbol{A}_{\mathrm{T}}^{(\mathrm{x})*} \times_4 \boldsymbol{A}_{\mathrm{T}}^{(\mathrm{y})*} \times_5 \boldsymbol{A}_{\mathrm{D}} \times_6 \boldsymbol{A}_{\mathrm{F}} \times_7 \boldsymbol{B}_{\mathrm{pol}}$$

Example: 64 x 32 URA MIMO, T=10, F= 128, 4 polarization pairs
 # coefficients: 10.485.760 → very large tensor!!

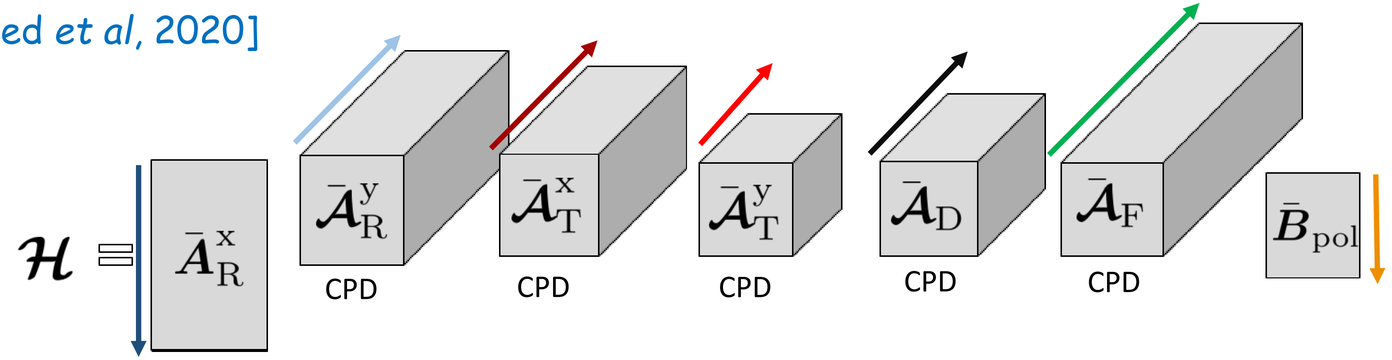
How to reduce complexity of channel representation and estimation ?

7D - channel tensor

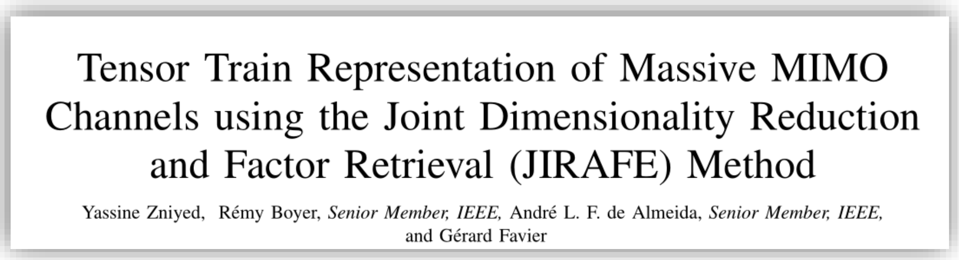


- Recasting the channel using Tensor Train model

[Znyed *et al*, 2020]

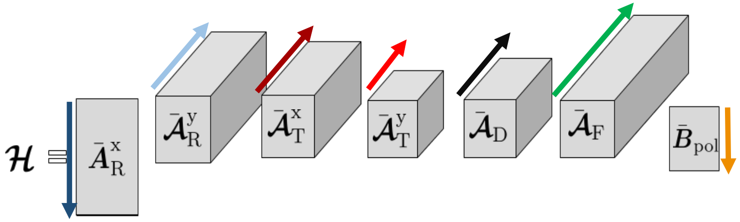


Tensor Train Based Channel Estimation (cont'd)



Tensor Train Representation of Massive MIMO Channels using the Joint Dimensionality Reduction and Factor Retrieval (JIRAFE) Method

Yassine Zniyed, Rémy Boyer, *Senior Member, IEEE,* André L. F. de Almeida, *Senior Member, IEEE,* and Gérard Favier



Dimensionality reduction

[Znyed *et al.*, 2020]

Tensor Train – SVD (TT-SVD)

$$[\bar{\boldsymbol{A}}_{\mathrm{R}}^{(\mathrm{x})}, \bar{\boldsymbol{\mathcal{A}}}_{\mathrm{R}}^{(\mathrm{y})}, \bar{\boldsymbol{\mathcal{A}}}_{\mathrm{T}}^{(\mathrm{x})}, \bar{\boldsymbol{\mathcal{A}}}_{\mathrm{T}}^{(\mathrm{y})}, \bar{\boldsymbol{\mathcal{A}}}_{\mathrm{D}}, \bar{\boldsymbol{\mathcal{A}}}_{\mathrm{F}}, \bar{\boldsymbol{B}}_{\mathrm{pol}}] \leftarrow \text{TT-SVD}(\boldsymbol{\mathcal{H}}, N_{\mathrm{p}})$$

Factors retrieval

Coupled LS optimization

$$F_{\text{global}} = \sum_{i=1}^{7} F_i$$

CPD's

$$F_1 = |||\bar{\boldsymbol{A}}_{\mathrm{R}}^{(\mathrm{x})} - \boldsymbol{A}_{\mathrm{R}}^{(\mathrm{x})} \boldsymbol{M}_1^{-1}|||_F^2$$

$$F_2 = |||\bar{\boldsymbol{\mathcal{A}}}_{\mathrm{R}}^{(\mathrm{y})} - \boldsymbol{\mathcal{I}}_{3,N_{\mathrm{p}}} \times_1 \boldsymbol{M}_1 \times_2 \boldsymbol{A}_{\mathrm{R}}^{(\mathrm{y})*} \times_3 \boldsymbol{M}_2^{-T}|||_F^2$$

$$F_3 = |||\bar{\boldsymbol{\mathcal{A}}}_{\mathrm{T}}^{(\mathrm{x})} - \boldsymbol{\mathcal{I}}_{3,N_{\mathrm{p}}} \times_1 \boldsymbol{M}_2 \times_2 \boldsymbol{A}_{\mathrm{T}}^{(\mathrm{x})*} \times_3 \boldsymbol{M}_3^{-T}|||_F^2$$

$$F_4 = |||\bar{\boldsymbol{\mathcal{A}}}_{\mathrm{T}}^{(\mathrm{y})} - \boldsymbol{\mathcal{I}}_{3,N_{\mathrm{p}}} \times_1 \boldsymbol{M}_3 \times_2 \boldsymbol{A}_{\mathrm{T}}^{(\mathrm{y})*} \times_3 \boldsymbol{M}_4^{-T}|||_F^2$$

$$F_5 = |||\bar{\boldsymbol{\mathcal{A}}}_{\mathrm{D}} - \boldsymbol{\mathcal{I}}_{3,N_{\mathrm{p}}} \times_1 \boldsymbol{M}_4 \times_2 \boldsymbol{A}_{\mathrm{D}} \times_3 \boldsymbol{M}_5^{-T}|||_F^2$$

$$F_6 = |||\bar{\boldsymbol{\mathcal{A}}}_{\mathrm{F}} - \boldsymbol{\mathcal{I}}_{3,N_{\mathrm{p}}} \times_1 \boldsymbol{M}_5 \times_2 \boldsymbol{A}_{\mathrm{F}} \times_3 \boldsymbol{M}_6^{-T}|||_F^2$$

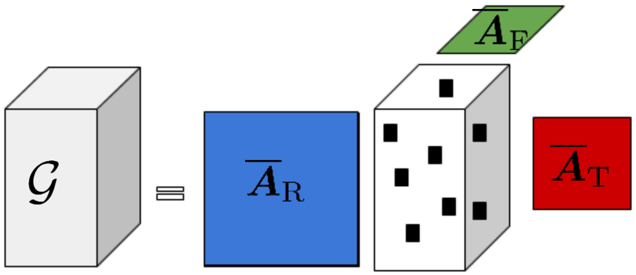
$$F_7 = |||\bar{\boldsymbol{B}}_{\mathrm{pol}} - \boldsymbol{M}_6 \boldsymbol{B}_{\mathrm{pol}}|||_F^2$$

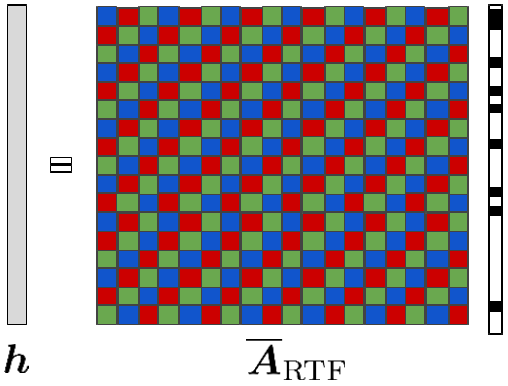
Sparse channel modeling & estimation

- Realistic channel models are not i.i.d →highly structured

- Algebraic channel structure is heterogeneous in different domains (e.g. space, frequency, time, polarization, etc...)

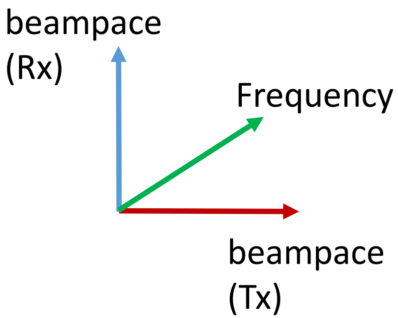
- Multidimensional channel structure is lost when working with vectorized (or “matricized”) versions of the channel

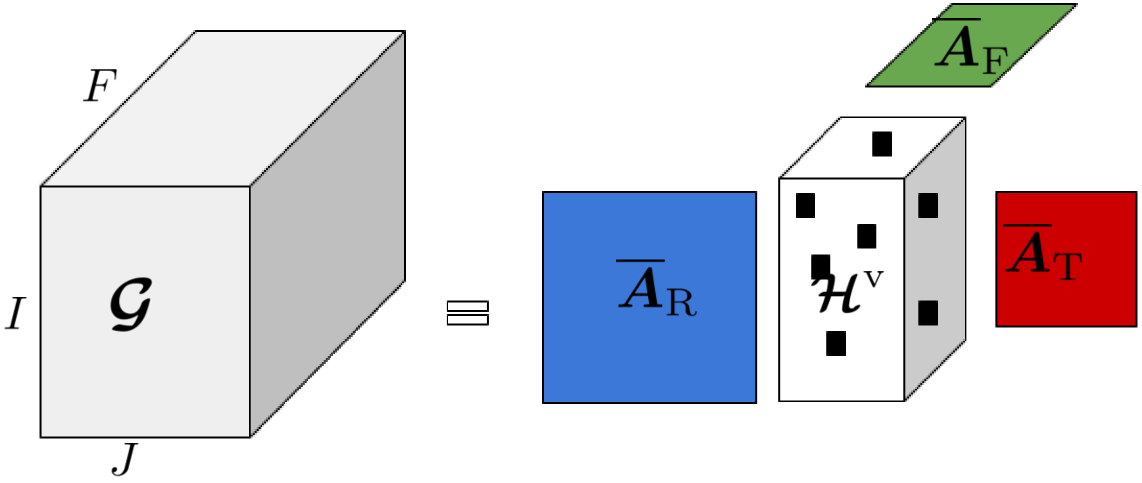




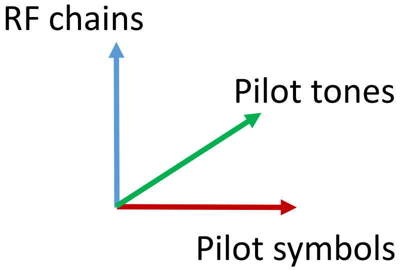
Sparse channel modeling & estimation (cont'd)

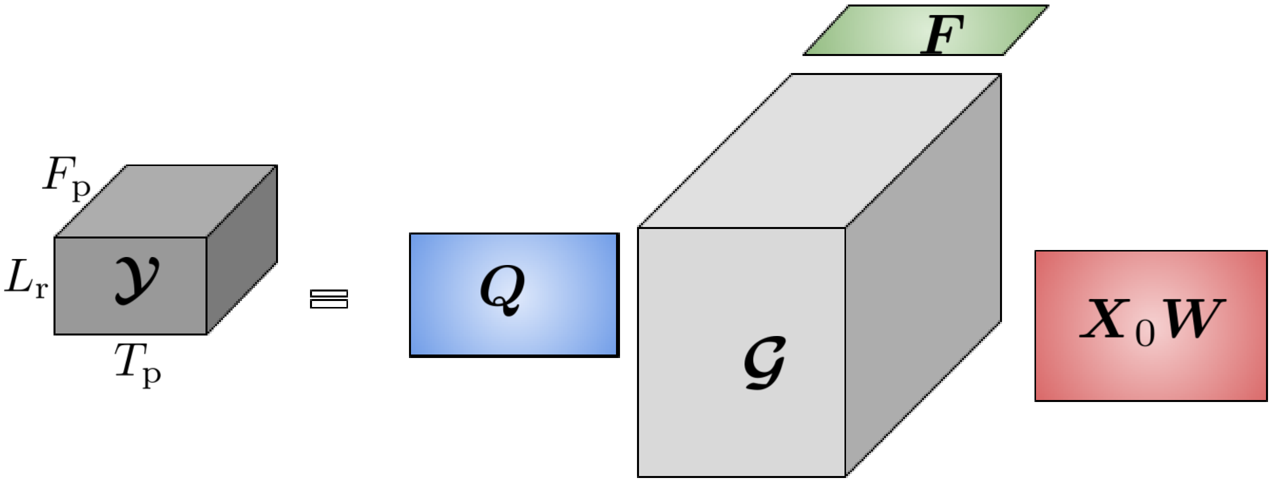
Channel tensor (sparse Tucker-3)





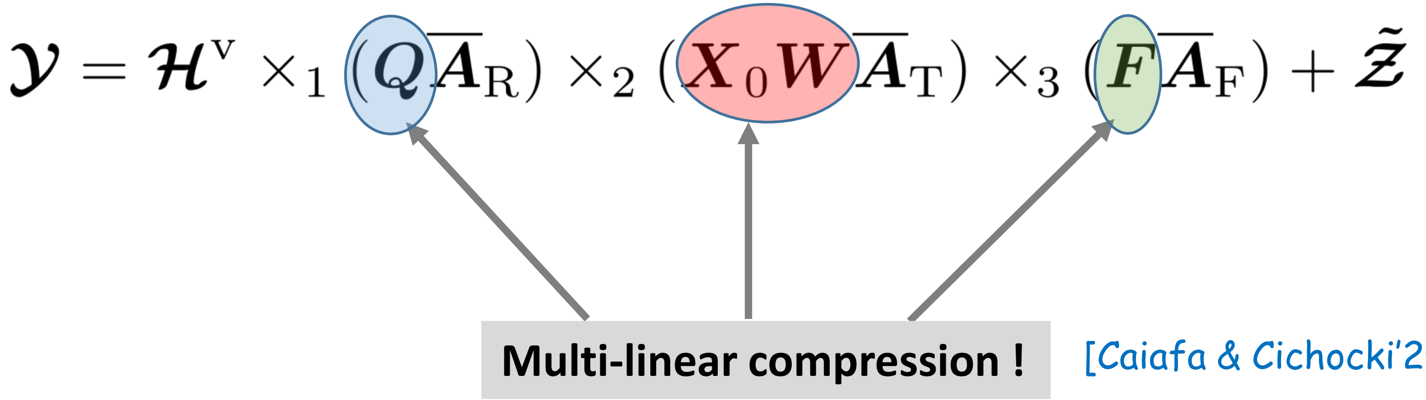
Compressed Rx signal tensor





Sparse channel modeling & estimation

- Expanding the 3D sparse channel tensor...



$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{H}}^{\mathrm{v}} \times_1 (\boldsymbol{Q}\overline{\boldsymbol{A}}_{\mathrm{R}}) \times_2 (\boldsymbol{X}_0\boldsymbol{W}\overline{\boldsymbol{A}}_{\mathrm{T}}) \times_3 (\boldsymbol{F}\overline{\boldsymbol{A}}_{\mathrm{F}}) + \tilde{\boldsymbol{\mathcal{Z}}}$$

Multi-linear compression !

[Caiafa & Cichocki'2013]

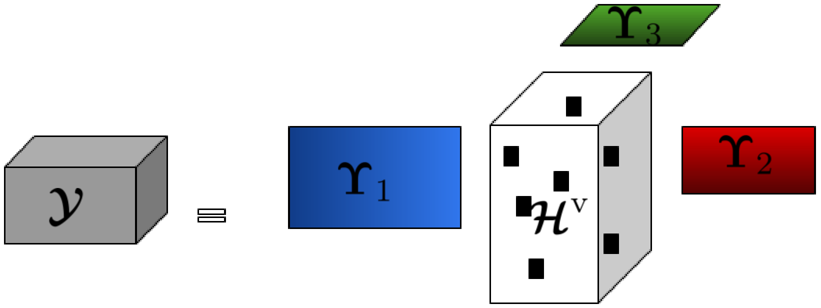
[Friedland, Li, Schonfeld '2014]

- Equivalent "vectorized" Kronecker- CS model [Duarte & Braniuk'2012]

$$\boldsymbol{y} = \big[(\boldsymbol{F}\overline{\boldsymbol{A}}_{\mathrm{F}}) \otimes (\boldsymbol{X}_0\boldsymbol{W}\overline{\boldsymbol{A}}_{\mathrm{T}}) \otimes (\boldsymbol{Q}\overline{\boldsymbol{A}}_{\mathrm{R}})\big]\boldsymbol{h}^{\mathrm{v}} + \tilde{\boldsymbol{z}}$$

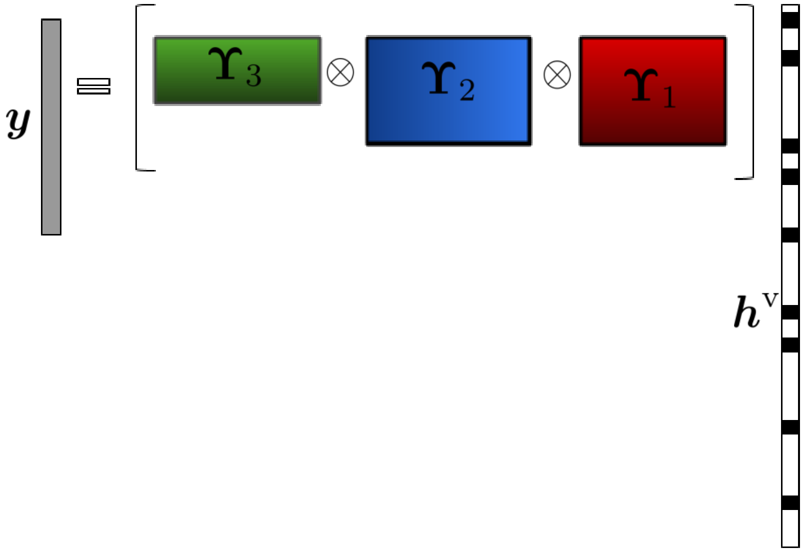
$$\boldsymbol{y} = \mathrm{vec}(\boldsymbol{\mathcal{Y}}), \quad \boldsymbol{h}^{\mathrm{v}} = \mathrm{vec}(\boldsymbol{\mathcal{H}}^{\mathrm{v}}), \quad \tilde{\boldsymbol{z}} = \mathrm{vec}(\tilde{\boldsymbol{\mathcal{Z}}})$$

Tensor-CS *vs.* Vector-CS



$$\mathrm{O}((L_{\mathrm{r}} + T_{\mathrm{p}} + F_{\mathrm{p}}))\log(L_{\mathrm{r}} + T_{\mathrm{p}} + F_{\mathrm{p}}) \leq C \leq \mathrm{O}(L_{\mathrm{r}}^3 + T_{\mathrm{p}}^3 + F_{\mathrm{p}}^3)$$

Complexity reduction



$$\mathrm{O}(L_{\mathrm{r}}T_{\mathrm{p}}F_{\mathrm{p}}\log(L_{\mathrm{r}}T_{\mathrm{p}}F_{\mathrm{p}}) \leq C \leq \mathrm{O}(L_{\mathrm{r}}^3T_{\mathrm{p}}^3F_{\mathrm{p}}^3)$$

Exploiting multilinearity + sparsity + low-rankness

- MIMO channel tensor w/ correlated scattering (angular spread)

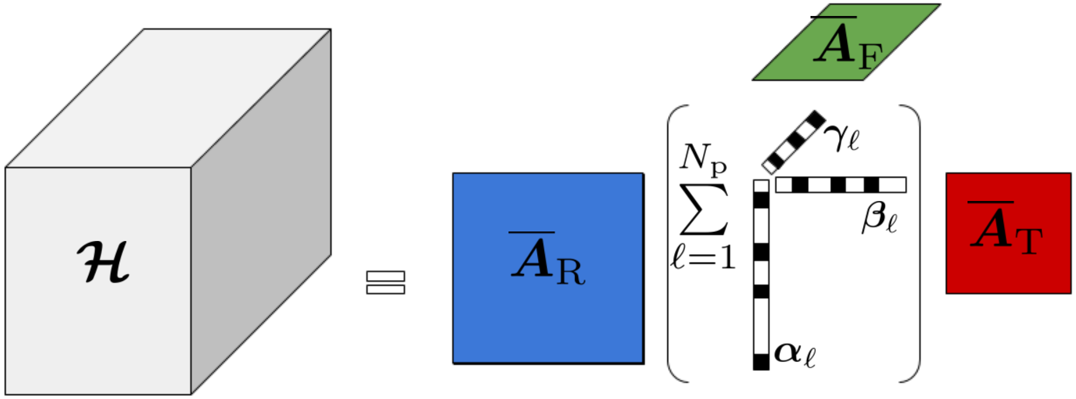
$$\boldsymbol{\mathcal{H}} = \sum_{\ell=1}^{N_\mathrm{p}} \left(\boldsymbol{A}_\mathrm{R}^{(\ell)}\boldsymbol{\alpha}_\ell\right) \circ \left(\boldsymbol{A}_\mathrm{T}^{(\ell)}\boldsymbol{\beta}_\ell\right) \circ \left(\boldsymbol{A}_\mathrm{F}^{(\ell)}\boldsymbol{\gamma}_\ell\right)$$

PARAFAC/CPD

$$= \left(\sum_{\ell=1}^{N_\mathrm{p}} \boldsymbol{\alpha}_\ell \circ \boldsymbol{\beta}_\ell \circ \boldsymbol{\gamma}_\ell\right) \times_1 \overline{\boldsymbol{A}}_\mathrm{R} \times_2 \overline{\boldsymbol{A}}_\mathrm{T} \times_3 \overline{\boldsymbol{A}}_\mathrm{F}$$

Sparse PARAFAC core

Basis matrices (dictionaries)



Tucker-3 model w/ sparse PARAFAC core

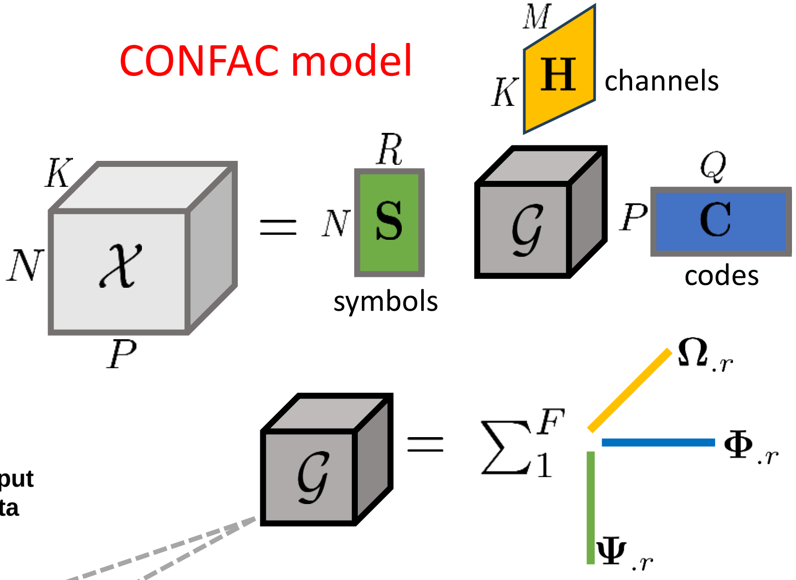
Design of semi-blind MIMO systems

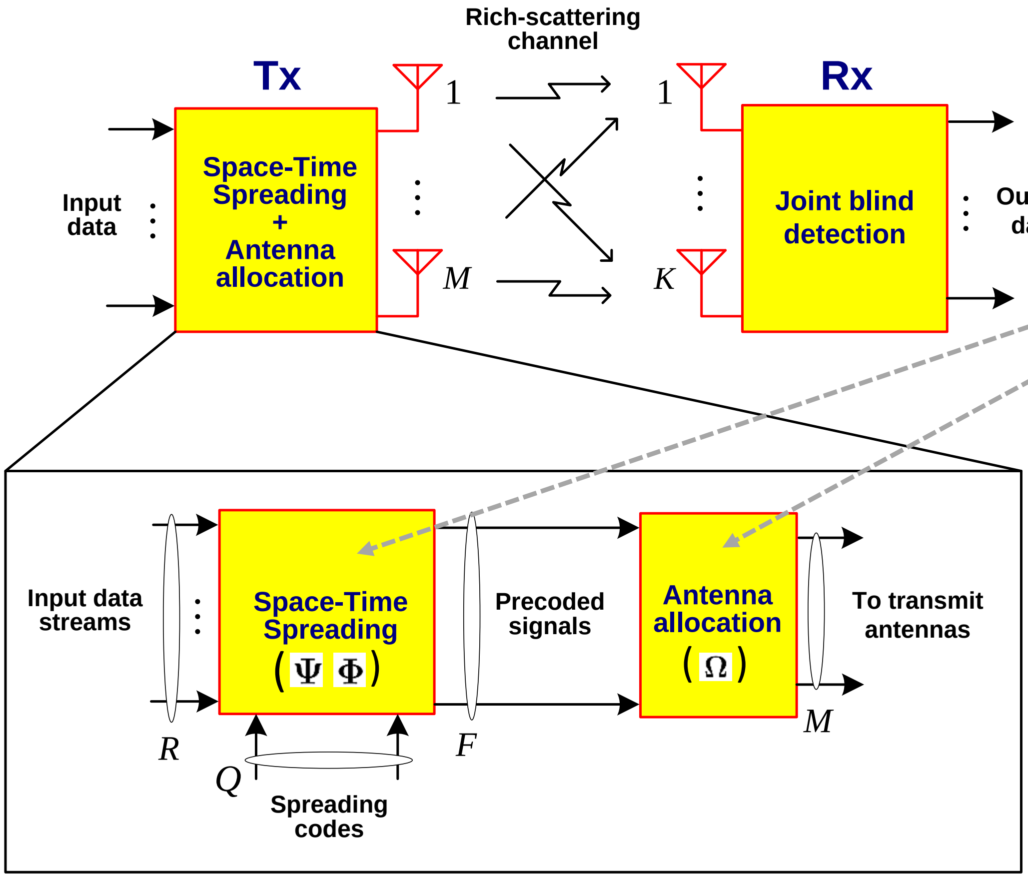
CONFAC based MIMO transceivers

Idea [de Almeida & Favier, 2008]

- Design **flexible space-time MIMO** schemes
- Capitalize on the **CONFAC uniqueness** to **jointly estimate channel and symbols** at the receiver

CONFAC model





$\mathbf{\Psi}$ ➡ *symbol allocation* (R x F)

$\mathbf{\Phi}$ ➡ *code allocation* (Q x F)

$\mathbf{\Omega}$ ➡ *antenna allocation* (M x F)

Designing tensor $\mathcal{G}$ defines the transmission scheme!

CONFAC-based MIMO system

Key features

- **Variable antenna allocation patterns:** Multiple data streams per transmit antenna
- **Variable spreading code reuse patterns:** Spreading codes can be reused by TX antennas
- **Transmission flexibility:** Several schemes possible by adjusting the allocation matrices

- Received signal (n-th symbol, p-th chip, k-th Rx antenna):

$$x_{k,n,p} = \sum_{m=1}^{M} \sum_{r=1}^{R} s_{n,r} c_{p,q} h_{k,m} \underbrace{g_{r,q,m}(\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega})}_{\textbf{Resource allocation tensor}}$$

with $F \geq \max(R, Q, M)$

PARAFAC DS-CDMA model

[Sidiropoulos et al, 2000]

$$\boldsymbol{\Psi} = \boldsymbol{\Phi} = \boldsymbol{\Omega} = \mathbf{I}_F$$
$$\mathcal{G}(\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega}) = \mathcal{I}_F$$

Note: columns of $\boldsymbol{\Psi}, \boldsymbol{\Phi},$ and $\boldsymbol{\Omega}$ are canonical basis vectors (1's and 0's)

Tensor Space-Time-Frequency (T-STF) Coding

Idea [de Almeida and Favier, 2014]

- Design generalized STF coding scheme with **allocation flexibility** over different STF domains (MIMO-OFDM-CDMA)

- T-STF coding model (5D)

$$x_{k,n,f,p,j} = \sum_{m=1}^{M}\sum_{r=1}^{R} \underset{\downarrow}{g_{m,r,f,p,j}}\, \underset{\downarrow}{h_{k,m,f}}\, \underset{\downarrow}{s_{n,r}}$$

Code tensor **Channel tensor** **Symbol matrix**

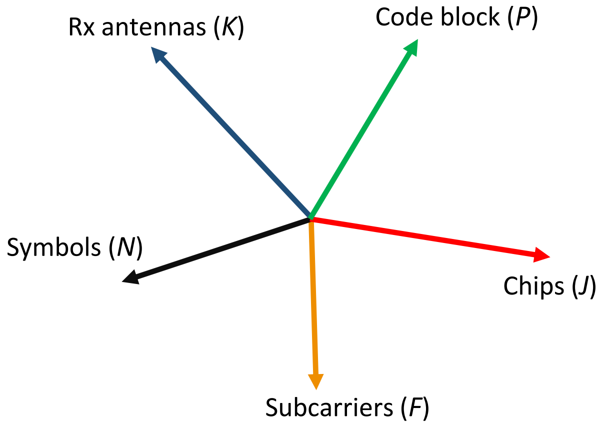
- Received signal (noiseless case)

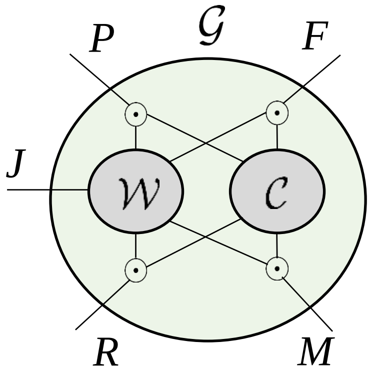
$$\mathcal{X} = \mathcal{G}\times_1\mathcal{H}\times_2\mathbf{S}$$

→ Tucker-(2-5) model

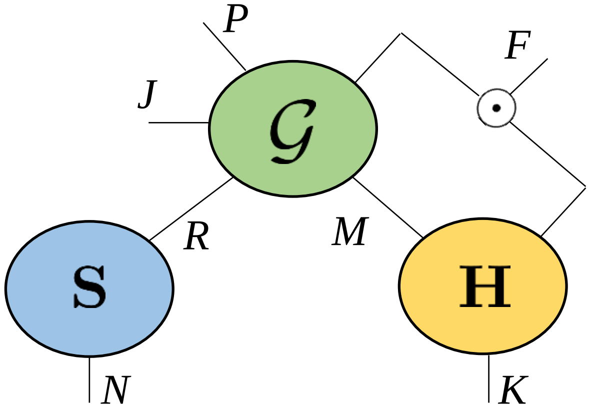
with $\mathcal{G} = \mathcal{W} \underset{\{m,r,f,p\}}{\odot} \mathcal{C}$

spreading tensor **allocation tensor**





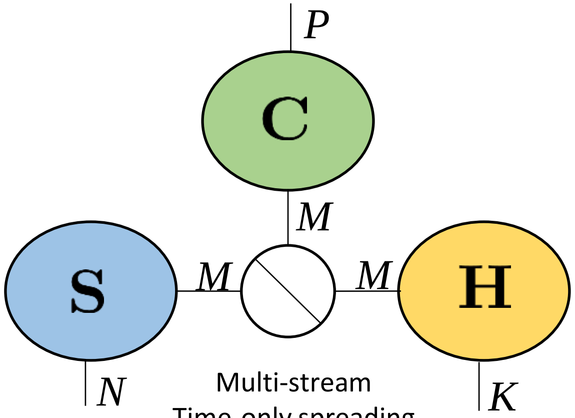
Coding tensor diagram



Received signal tensor diagram

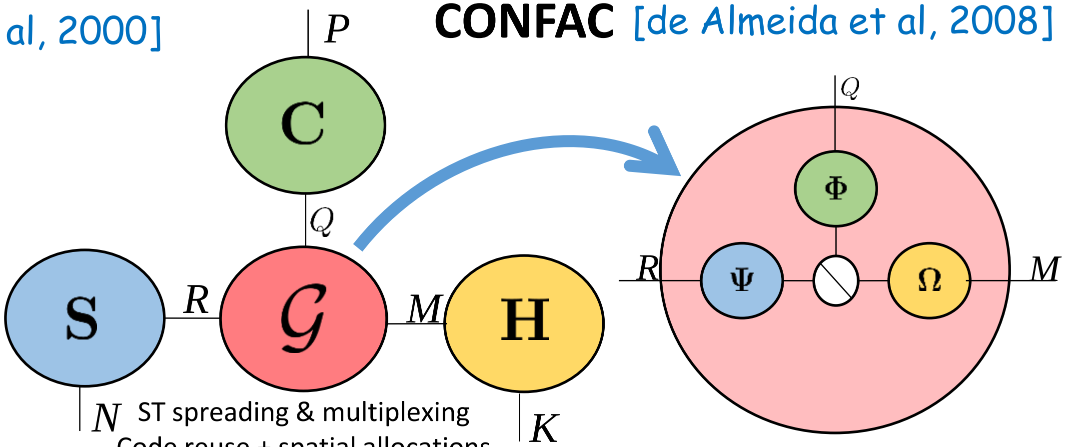
T-STF vs. CONFAC vs. PARAFAC schemes

PARAFAC [Sidiropoulos et al, 2000]



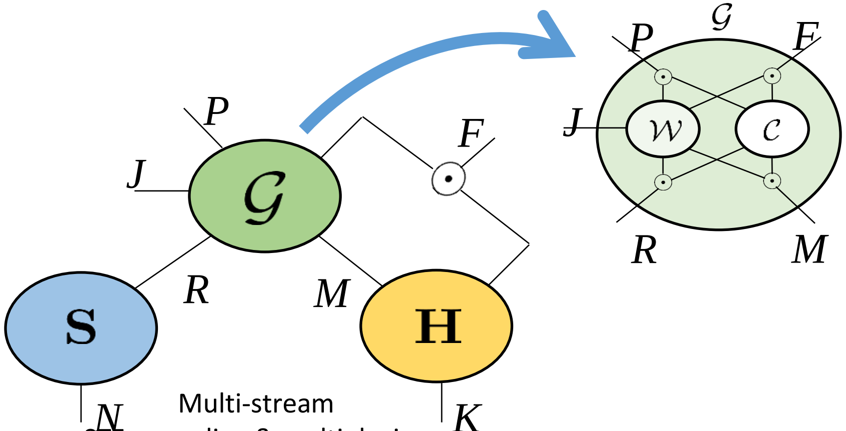
Multi-stream
Time-only spreading

CONFAC [de Almeida et al, 2008]



ST spreading & multiplexing
Code reuse + spatial allocations

T-STF [de Almeida & Favier, 2014]



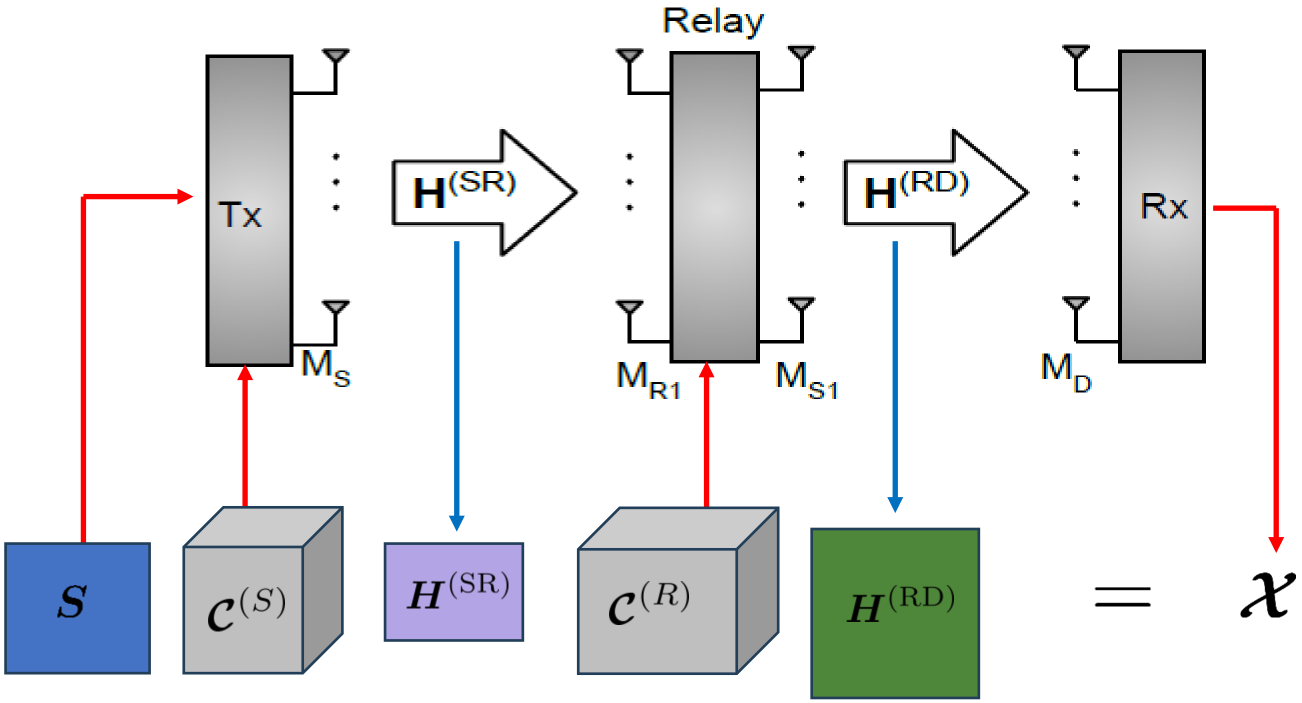
Multi-stream
STF spreading & multiplexing
Full allocations

MIMO Relay Systems

Semi-Blind MIMO Relay Systems

Idea: Use tensor coding at source and relay to jointly estimate the involved channels (source-relay and relay-destination)

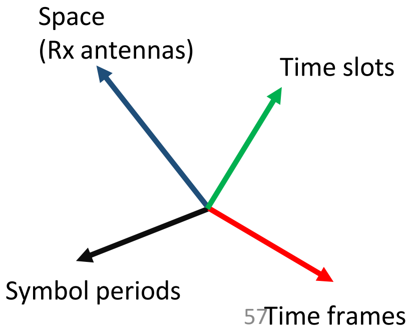
[Ximenes et al, 2015]
[Fernandes et al, 2016]
[Znyed et al, 2018]
[Sokal et al, 2020]



$$\boldsymbol{\mathcal{X}} \in \mathbb{C}^{T \times P \times J \times M_R}$$

Nested Tucker-(2,4) model

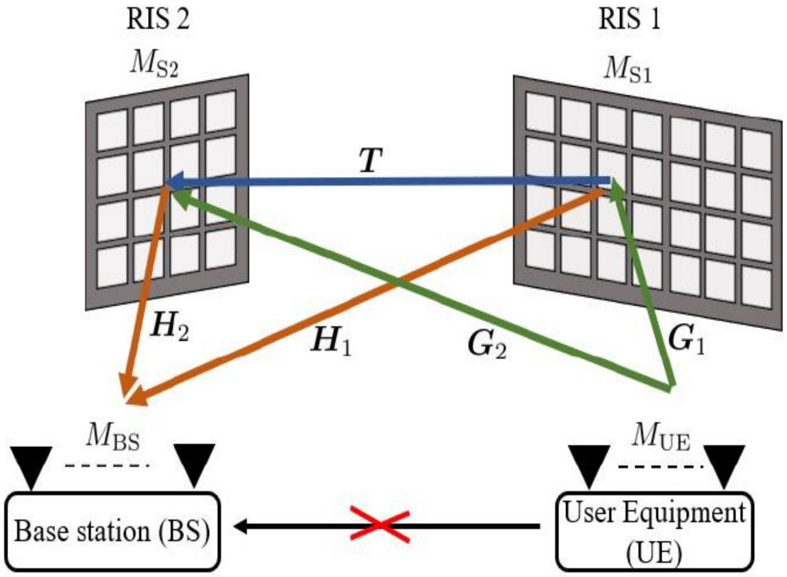
$$\mathcal{X} = (\mathcal{C}^{(S)} \times_2^1 \mathbf{H}^{(SR)} \times_2^1 \mathcal{C}^{(R)}) \times_1 \mathbf{S} \times_2 \mathbf{H}^{(RD)}$$



Reconfigurable Intelligent Surfaces

Channel estimation with Reconfigurable Surfaces

Problem: Jointly estimate multiple channels in a communication system aided by **reconfigurable surfaces** [de Almeida et al, 2024]



Single reflection links (PARAFAC):

$$\mathcal{Y}_{\mathrm{RIS}_1} = \mathcal{I}_{3,M_{S1}} \times_1 \boldsymbol{H}_1 \times_2 \boldsymbol{G}_1^{\mathrm{T}} \times_3 \boldsymbol{\Theta}_1$$

and

$$\mathcal{Y}_{\mathrm{RIS}_2} = \mathcal{I}_{3,M_{S1}} \times_1 \boldsymbol{H}_2 \times_2 \boldsymbol{G}_2^{\mathrm{T}} \times_3 \boldsymbol{\Theta}_2$$

Double reflection links (Nested PARAFAC):

$$\mathcal{Y}^{(1)}_{\mathrm{RIS}_{12}} = \mathcal{I}_{3,M_{S2}} \times_1 \boldsymbol{H}_2 \times_2 [\boldsymbol{\Theta}_1 \diamond \boldsymbol{G}_1^{\mathrm{T}}]\boldsymbol{T}^{\mathrm{T}} \times_3 \boldsymbol{\Theta}_2$$

or

$$\mathcal{Y}^{(2)}_{\mathrm{RIS}_{12}} = \mathcal{I}_{3,M_{S1}} \times_1 [\boldsymbol{\Theta}_2 \diamond \boldsymbol{H}_2]\boldsymbol{T} \times_2 \boldsymbol{G}_1^{\mathrm{T}} \times_3 \boldsymbol{\Theta}_1$$

Combine $\mathcal{Y}_{\mathrm{RIS}_1}$ and $\mathcal{Y}^{(2)}_{\mathrm{RIS}_{12}}$ to estimate $\boldsymbol{G}_1$

Combine $\mathcal{Y}_{\mathrm{RIS}_2}$ and $\mathcal{Y}^{(1)}_{\mathrm{RIS}_{12}}$ to estimate $\boldsymbol{H}_2$

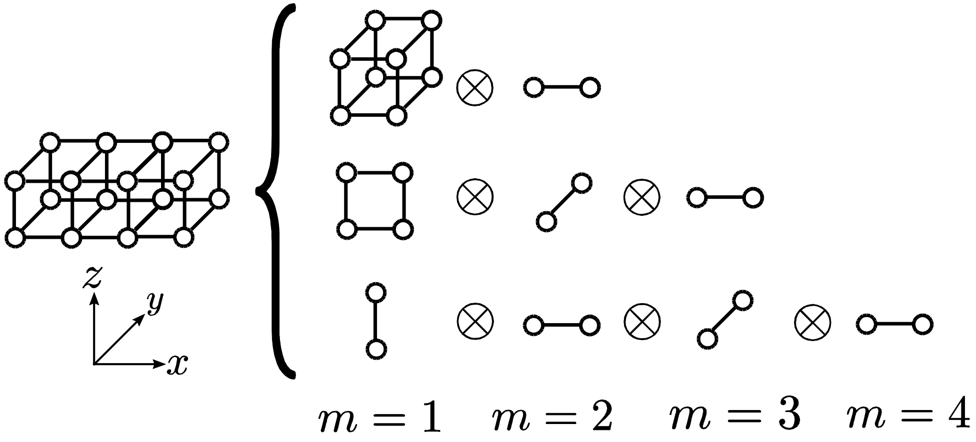
→ Coupled Nested PARFAFAC decomp.

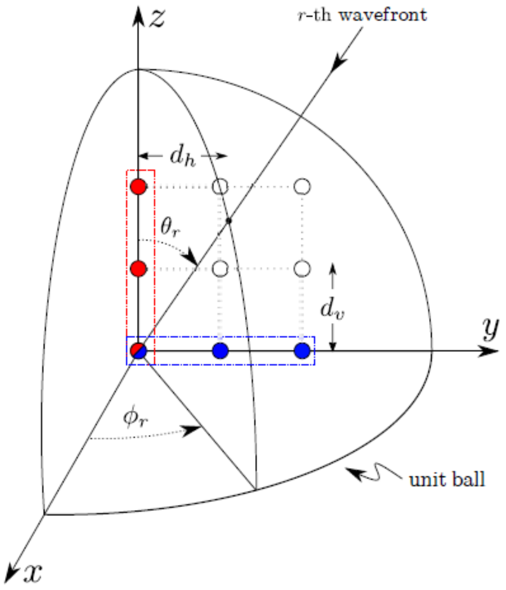
Multi-Linear Beamforming

Why multi-linear beamforming?

- As the size of a sensor array grows, the beamforming operation needs more...
  - ❖ Samples to estimate statistics
  - ❖ Computation time to obtain weights

- **Idea**: Exploit the algebraic structure of separable arrays → multi-linearity property





Multi-linear filtering

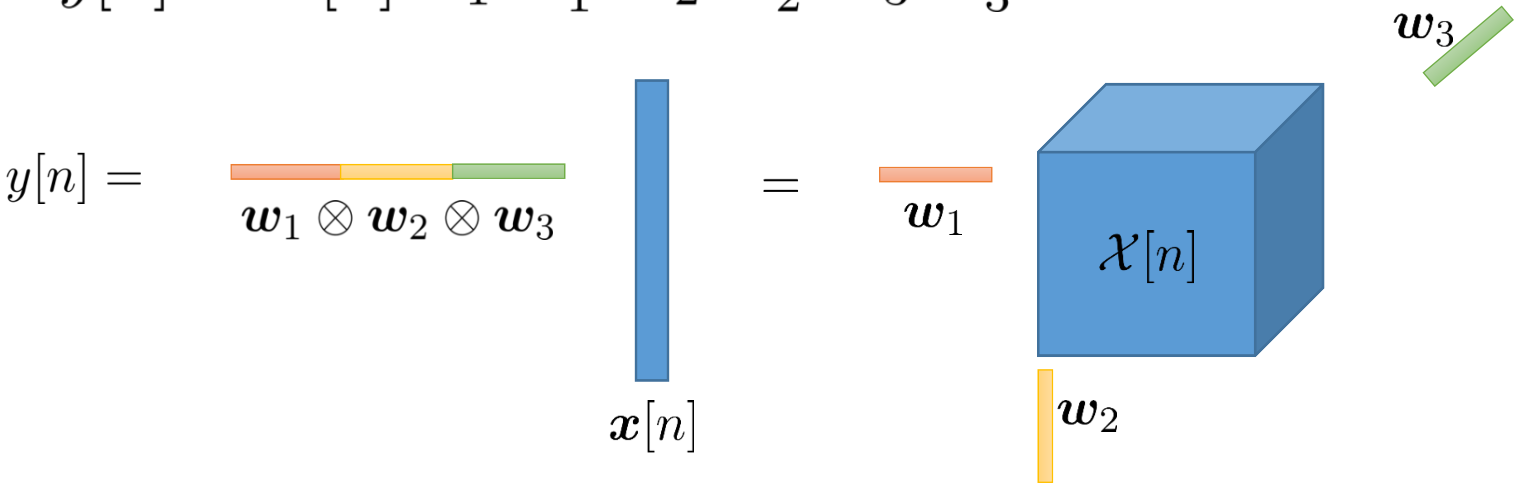
Idea: Kronecker filters as multilinear maps

- Consider the trilinear filter:

$$y[n] = \boldsymbol{w}^{\mathsf{H}}\boldsymbol{x}[n] = (\boldsymbol{w}_1 \otimes \boldsymbol{w}_2 \otimes \boldsymbol{w}_3)^{\mathsf{H}}\boldsymbol{x}[n]$$

- Reshape the input signal vector into a 3d tensor:

$$y[n] = \mathcal{X}[n] \times_1 \boldsymbol{w}_1^{\mathsf{H}} \times_2 \boldsymbol{w}_2^{\mathsf{H}} \times_3 \boldsymbol{w}_3^{\mathsf{H}}$$



Multi-linear filtering (cont'd)

- From tensor algebra, the trilinear filter output can be written as

$$\begin{aligned} y[n] &= \boldsymbol{w}_1^{\mathsf{H}} \boldsymbol{X}_{(1)}[n](\boldsymbol{w}_3 \otimes \boldsymbol{w}_2)^* = \boldsymbol{w}_1^{\mathsf{H}} \boldsymbol{u}_1[n] \\ &= \boldsymbol{w}_2^{\mathsf{H}} \boldsymbol{X}_{(2)}[n](\boldsymbol{w}_3 \otimes \boldsymbol{w}_1)^* = \boldsymbol{w}_2^{\mathsf{H}} \boldsymbol{u}_2[n] \\ &= \boldsymbol{w}_3^{\mathsf{H}} \boldsymbol{X}_{(3)}[n](\boldsymbol{w}_2 \otimes \boldsymbol{w}_1)^* = \boldsymbol{w}_3^{\mathsf{H}} \boldsymbol{u}_3[n] \end{aligned}$$

Keep fixed

Linear w.r.t. each subfilter

Idea:

- Design each "subfilter" instead of full filter
- Computational complexity reduction

Tensor beamforming algorithms

- Alternating optimization approaches
  - ❖ Tensor LMS [Rupp & Schwarz’2015]
  - ❖ Tensor GSC [Miranda et al’2015]
  - ❖ **Tensor MMSE** [Ribeiro et al’2016, Ribeiro et al’2019]
  - ❖ **Tensor LCMV** [Ribeiro et al’2019]
  - ❖ **Tensor Frost** [Ribeiro et al’2019]

N-dimensional filter with $N = N_1 N_2 \, N_3$

- Example: Trilinear filter design $\boldsymbol{w} = \underset{N_1}{\boldsymbol{w}_1} \otimes \underset{N_2}{\boldsymbol{w}_2} \otimes \underset{N_3}{\boldsymbol{w}_3}$

1. Random initialization for $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$
2. Optimize for $\boldsymbol{w}_1$ with $\boldsymbol{w}_2, \boldsymbol{w}_3$ fixed – $O(N_1^3)$ multiplications
3. Optimize for $\boldsymbol{w}_2$ with $\boldsymbol{w}_1, \boldsymbol{w}_3$ fixed – $O(N_2^3)$ multiplications
4. Optimize for $\boldsymbol{w}_3$ with $\boldsymbol{w}_1, \boldsymbol{w}_2$ fixed – $O(N_3^3)$ multiplications
5. Has converged? If not, go back to step 2

$O(N_1^3 + N_2^3 + N_3^3)$ vs. $O(N^3)$

Each filter is updated with alternating optimization methods

Multi-linear Constellation Designs

Multi-linear constellation design

Principle

Any *M*-PSK constellation can be factorized into $P \leq \log_2 M$ different constellation sets:

$$\Phi = \Phi_0 \otimes \Phi_1 \cdots \otimes \Phi_{P-1}$$

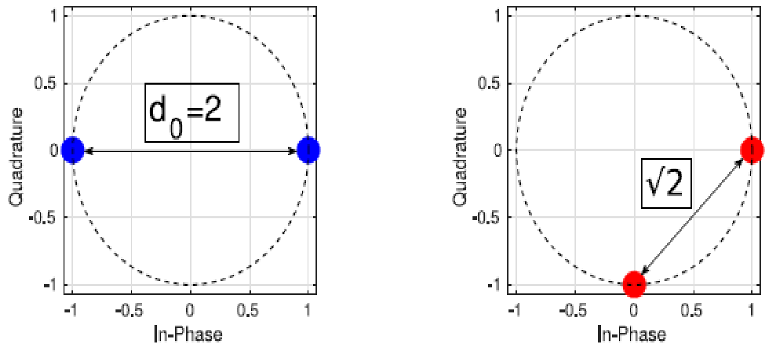
Signal model

$$\boldsymbol{y}[k] = h[k]\boldsymbol{x}[k] + \boldsymbol{n}[k]$$

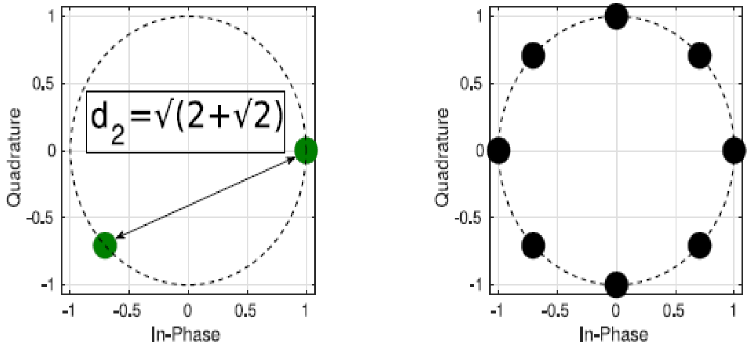
with

$$\boldsymbol{x}[k] = \boldsymbol{s}_N[k] \otimes \cdots \otimes \boldsymbol{s}_1[k]$$

Multi-linear *M*-PSK constellation



(a) $\Phi_0 \in \text{BPSK}$ (b) Φ_1

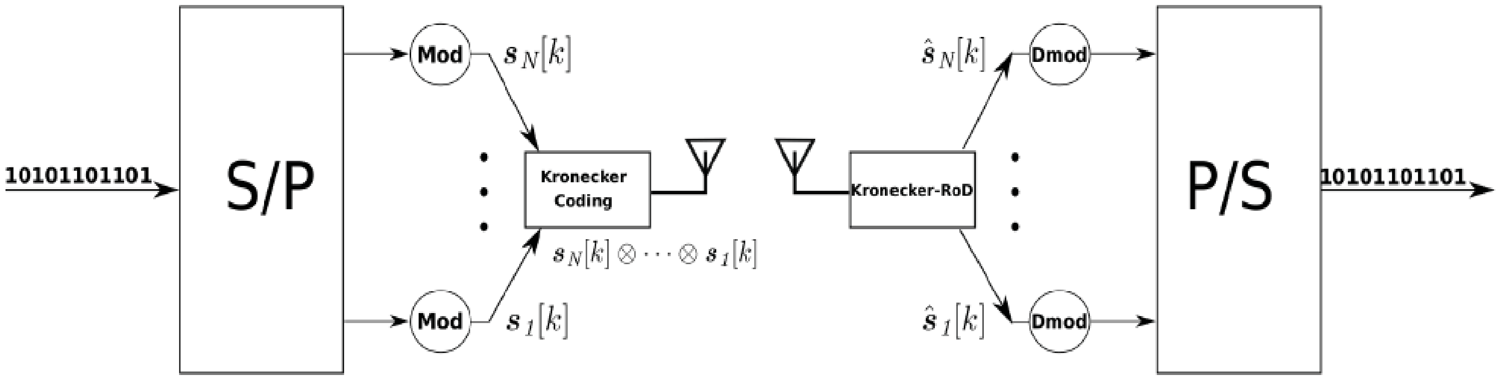


(c) Φ_2 (d) $\Phi = \Phi_0 \otimes \Phi_1 \otimes \Phi_2$

Universidade Federal do Ceará

Multi-linear constellation design

Transceiver



- Received signal after matched filtering (MF)

$$\hat{\boldsymbol{y}}[k] = h^*[k]\boldsymbol{y}[k]$$

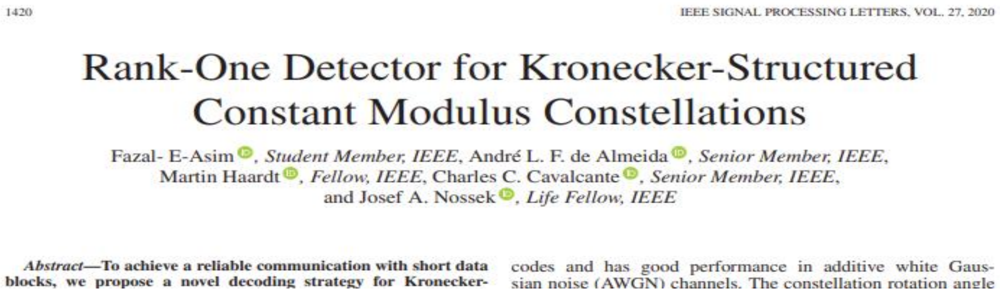
- Decoding as *N*-th order rank-one tensor approx. problem

$$\min_{\boldsymbol{s}_1,\ldots,\boldsymbol{s}_N} \left\| \hat{\mathcal{Y}} - \boldsymbol{s}_1 \circ \cdots \circ \boldsymbol{s}_N \right\|_F^2$$

- Equivalent solution: maximize the tensor Rayleigh quotient

$$T(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N) = \frac{\left|(\boldsymbol{s}_N \otimes \cdots \otimes \boldsymbol{s}_1)^T vec(\hat{\mathcal{Y}})\right|}{\|\boldsymbol{s}_1\|_2 \ldots \|\boldsymbol{s}_N\|_2}$$

Multi-linear constellation design



1420 IEEE SIGNAL PROCESSING LETTERS, VOL. 27, 2020

Rank-One Detector for Kronecker-Structured Constant Modulus Constellations

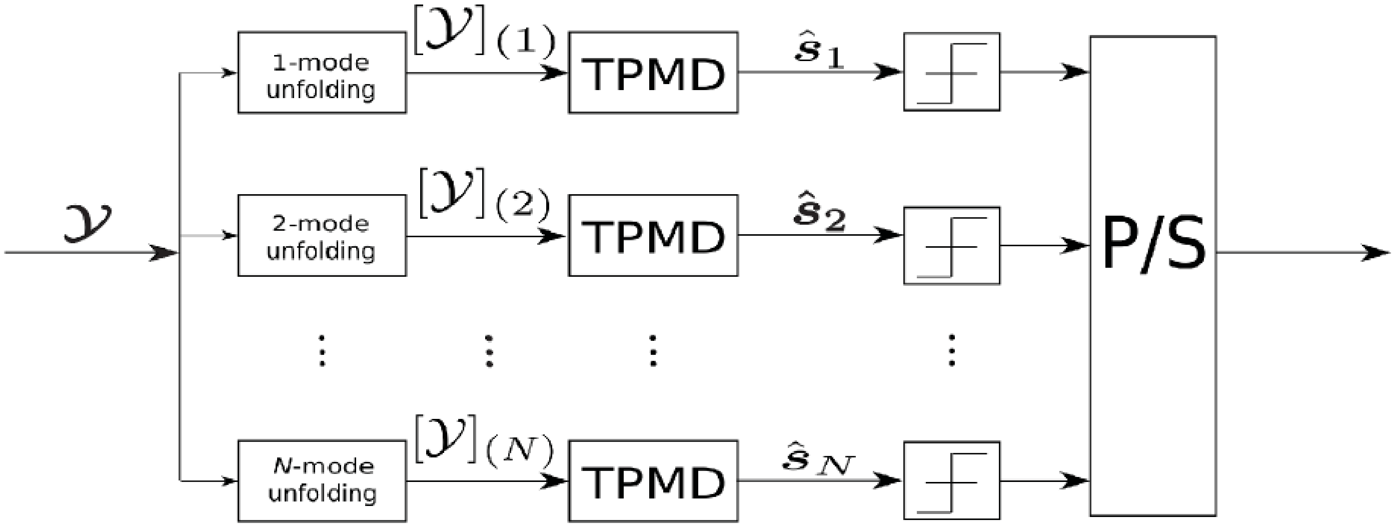
Fazal- E-Asim, *Student Member, IEEE*, André L. F. de Almeida, *Senior Member, IEEE*,
Martin Haardt, *Fellow, IEEE*, Charles C. Cavalcante, *Senior Member, IEEE*,
and Josef A. Nossek, *Life Fellow, IEEE*

***Abstract*—To achieve a reliable communication with short data blocks, we propose a novel decoding strategy for Kronecker-**

codes and has good performance in additive white Gaussian noise (AWGN) channels. The constellation rotation angle

Receiver processing

Kronecker Rank-One Detector (Kronecker-RoD)



Note: Decoding can be parallelized → reduced latency

Q&A

Thank you!

✉ andre@gtel.ufc.br
Federal University of Ceará
profalmeida.com
Fortaleza, Brazil