Tensor Decompositions: Models, Applications, Algorithms, Uniqueness

Dimitri Nion

Post-Doc fellow, KU Leuven,

Kortrijk, Belgium

E-mail: Dimitri.Nion@kuleuven-kortrijk.be

Homepage: http://perso-etis.ensea.fr/~nion/

I3S, Sophia-Antipolis, December 11th 2008

Preliminary

Tensor Decompositions

Q: What is this ?

R: Powerful **multi-linear algebra** tools that generalize matrix decompositions.

Q: Where are they useful ?

R: Increasing number of applications involve manipulation of multi-way data, rather than 2-way data.

Q: How powerful are they compared to matrix decompositions?

R: Uniqueness properties + Better exploitation of the multi-dimensional nature of data

Key research axes:

→ Development of new models/decompositions

→ Development of algorithms to compute decompositions

→ Uniqueness bounds of tensor decompositions

→ New applications, or existing applications where the multi-way nature of data was ignored until now

Roadmap

I. Introduction

II. A few Tensor Decompositions:
PARAFAC, HOSVD/Tucker, Block-Decompositions

III. Algorithms to compute Tensor Decompositions

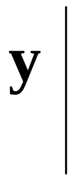
IV. Applications

V. Conclusion and Future Research

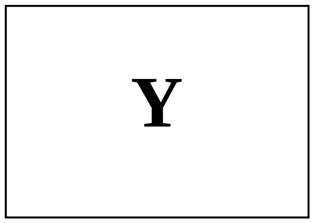
What is a tensor ?

Tensor of order N = Array with N dimensions

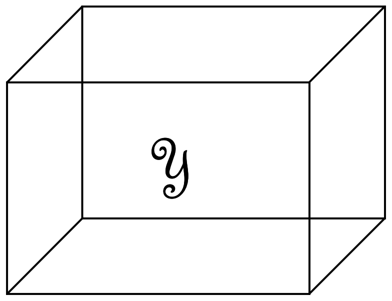
For N>2, « Higher-Order Tensors »



= 1st-order tensor



= 2nd-order tensor



= 3rd-order tensor

Multi-Way Processing, why?

General motivation for using tensor signal representation and processing :

« **If by nature, a signal is multi-dimensional, then its tensor representation allows to use multilinear algebra tools, which are more powerful than linear algebra tools.** »

Many signals are tensors :

- (R,G,B) image can be represented as a tensor

- Video sequence is a tensor of consecutive frames

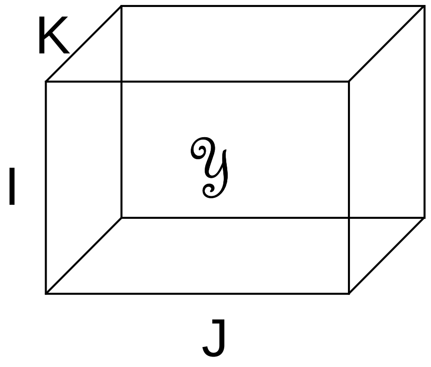
- Multi-variate signals, varying e.g. with time, temperature, illumination, sensor positions, etc…

Tensor models: an increasing number of applications

Various disciplines:

- Phonetics
- Psychometry
- Chemometrics (spectroscopy, chromatography)
- Image and video compression and analysis
- Scientific programming
- Sensor analysis
- Multi-Way Principal Component Analysis (PCA)
- Blind Source Separation and Independent Component Analysis (ICA)
- Telecommunications (wireless communications)

Multi-Way Data



→ Set of K matrices of size IxJ

↓

One matrix observed K times
(ex: K = time, K = number of sensors, etc)

→ 3-way tensor (« third-order tensor »)

Multiple variables → extension to N-way tensors

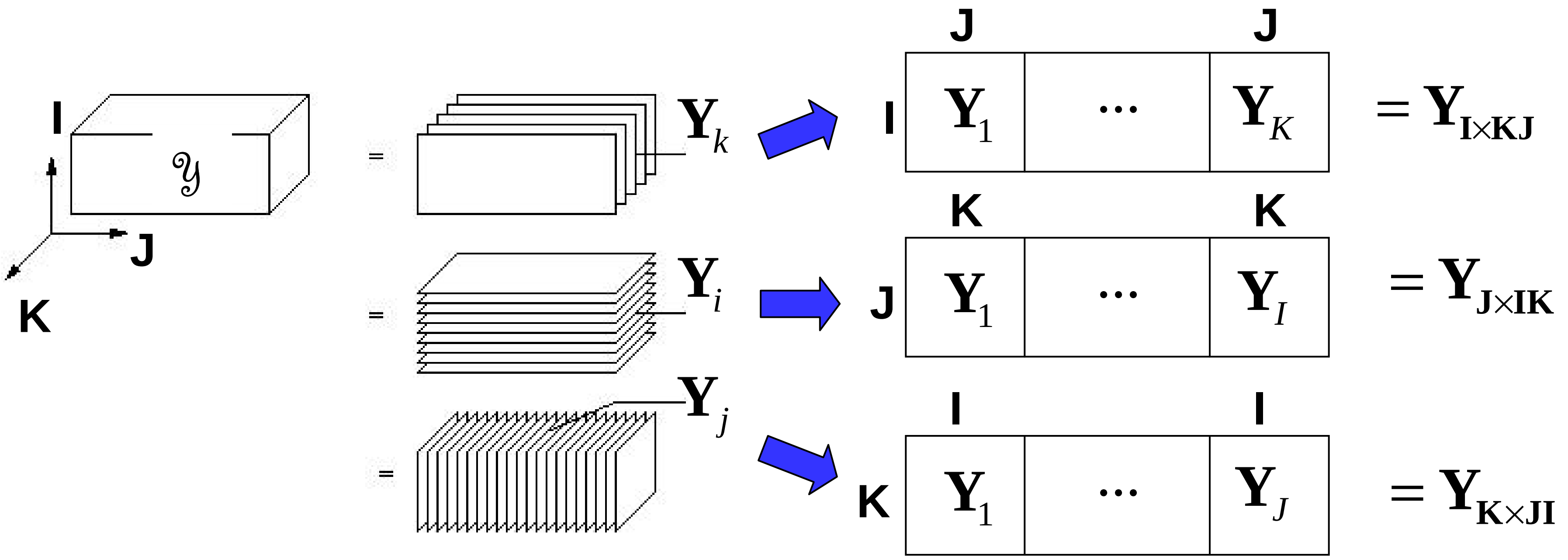
How to perform Multi-Way Analysis?

- Via tensor-algebra tools (=multilinear algebra tools)

- Matrix tools (SVD, EVD, QR, LU) have to be generalized

→**Tensor Decompositions**

Tensor Unfolding (“matricization”)



Multi-Way Analysis?

- One can choose one matrix representation of $\mathcal{Y}$ and apply matrix tools (ex: matrix SVD for Principal Component Analysis (PCA))

- **Problem:** the multi-way structure is then ignored

- **Feature of N-way analysis:** exploit the N matrices simultaneously

Roadmap

I. Introduction

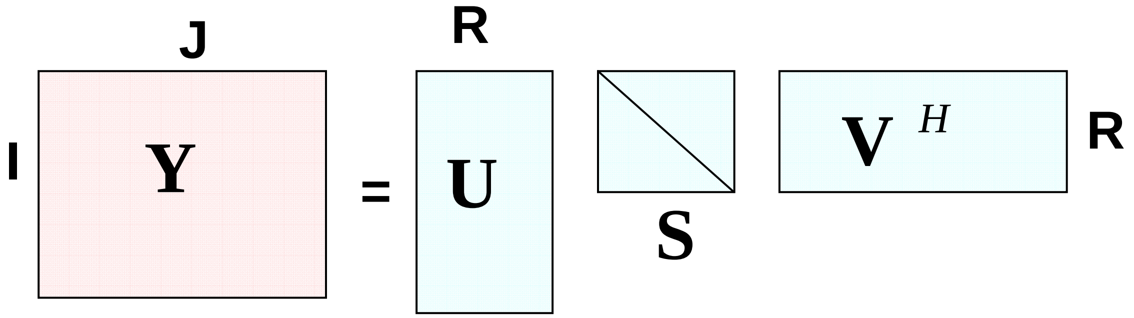
II. A few Tensor Decompositions:
PARAFAC, HOSVD/Tucker, Block-Decompositions

III. Algorithms to compute Tensor Decompositions

IV. Applications

V. Conclusion and Future Research

Matrix Singular Value Decomposition (SVD)



$$\begin{cases} \mathbf{U}^H\mathbf{U} = \mathbf{I} \text{ and } \mathbf{V}^H\mathbf{V} = \mathbf{I} \\ \mathbf{S} = diag(\sigma_1, \ldots, \sigma_R) \end{cases}$$

→ unitary matrices

→ Singular values in decreasing order

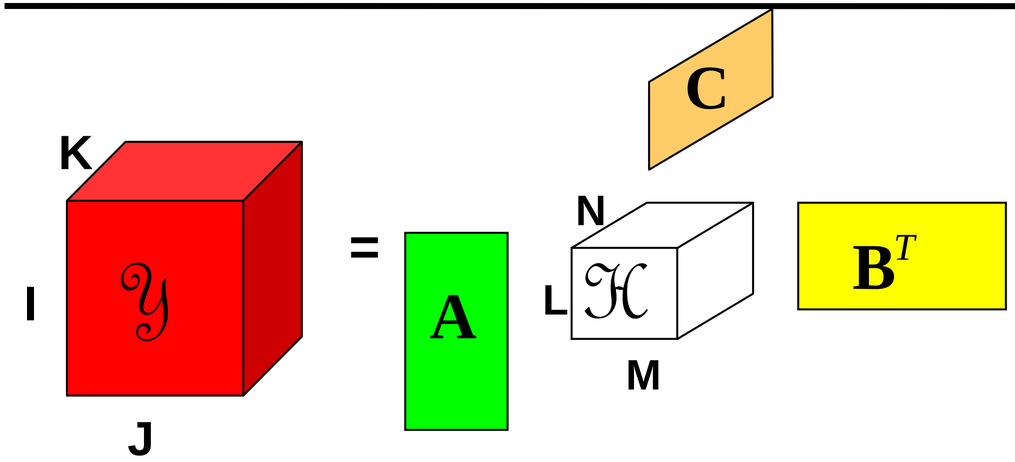
If *rank*(**Y**)>R, this truncated SVD is the best rank-R approx. of **Y**

In general a matrix factorization **Y=UV**$^{\mathbf{H}}$ is *not* unique:

$$\mathbf{Y=UV^H=UPP^{-1}V^H}$$

The SVD is unique because of unitary constraints on **U** and **V** and ordering constraint of the singular values in **S**

Tucker-3 Decomposition [Tucker 1966]



$$y_{ijk} = \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{n=1}^{N} a_{il} b_{jm} c_{kn} h_{lmn}$$

$$\mathcal{Y} = \mathcal{H} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

- Tucker-3 = 3-way PCA. One unitary base (A, B, C) per mode (Tucker-1, Tucker-2,…, Tucker-N are possible).

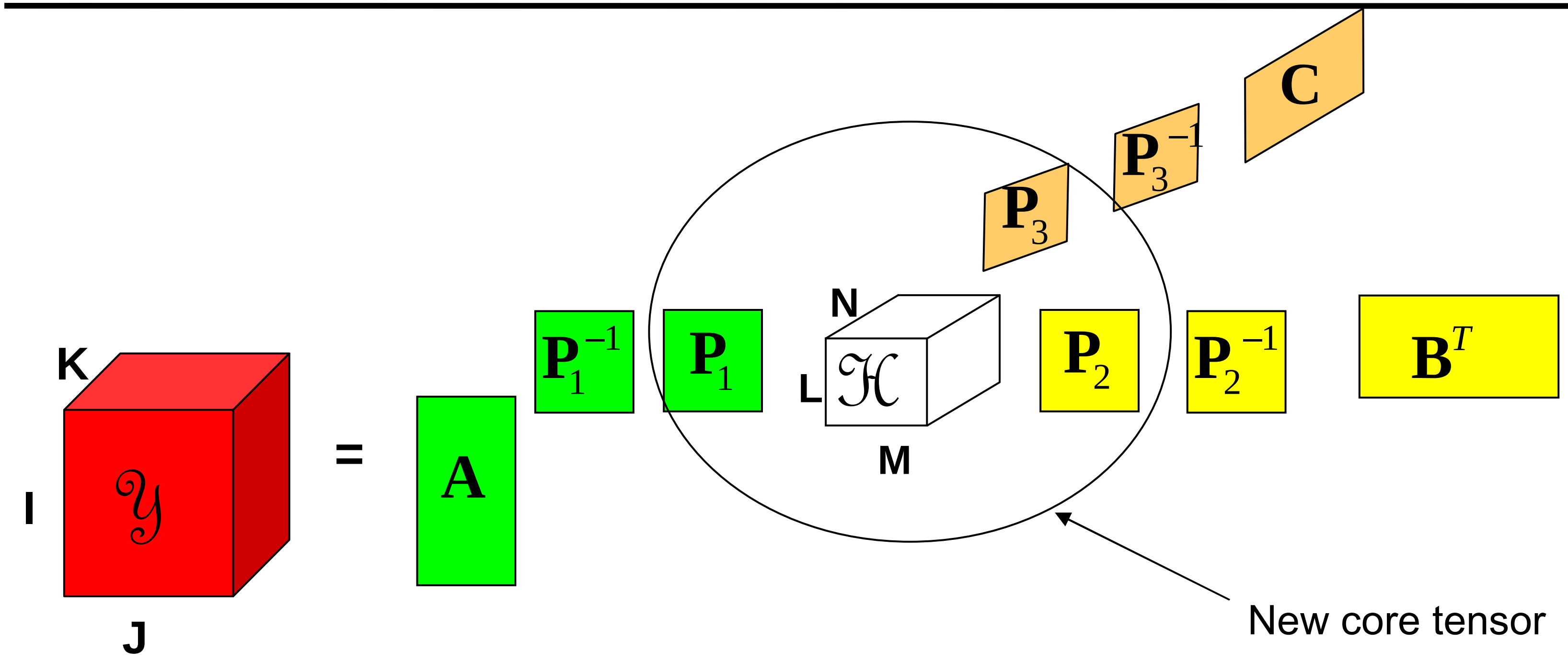
- If A, B, C are unitary matrices, TUCKER=HOSVD (« Higher Order Singular Value Decomposition »)

- $\mathcal{H}$ is the representation of $\mathcal{Y}$ in the reduced spaces.

- The number of principal components may be different in the three modes i.e. $L \neq M \neq N$

- $\mathcal{H}$ is not diagonal (difference with matrix SVD).

Uniqueness of Tucker-3 Decomposition



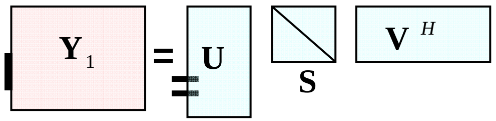
New core tensor

- Tucker not unique: rotational freedom in each mode.

→ **A**, **B**, **C** are not unique (only subspace estimates).

The best rank-(L,M,N) approximation [De Lathauwer, 2000]

Y_1 = truncated Matrix SVD of Y

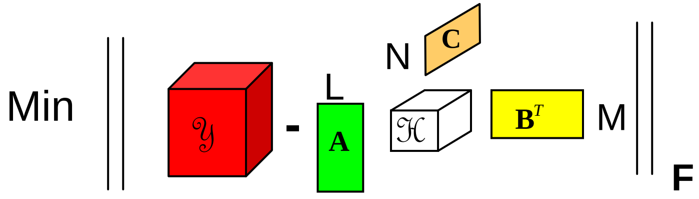


$\mathbf{Y_1}$ is the best lower rank approximation of **Y** (in the Frobenius norm sense):

$$\text{Min } \|\mathbf{Y}-\mathbf{Y_1}\|_F$$

$$\text{s.t. } \mathbf{Y_1} \text{ is rank-R}$$

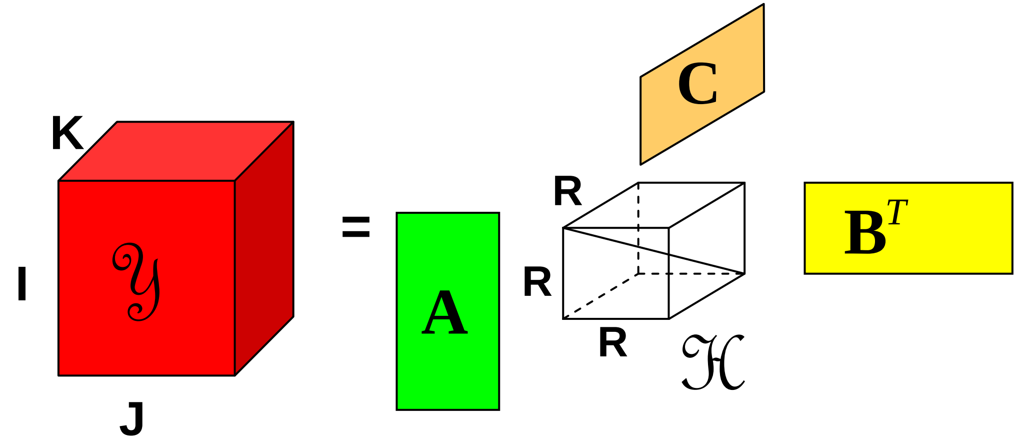
Question: Is the truncated HOSVD, the best rank-(L,M,N) approximation of $\mathcal{Y}$ **?** **NO**



The truncated HOSVD is only a good rank-(L,M,N) approximation of $\mathcal{Y}$.

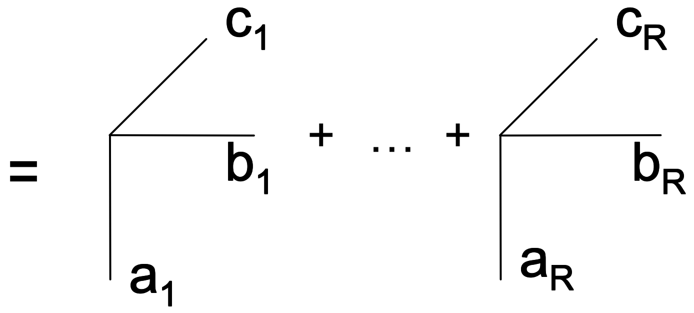
To find the best one, one usually starts with the truncated HOSVD (initialization) and then alternate updates of the 3 subspace matrices **A**, **B** and **C**.

PARAFAC Decomposition [Harshman 1970]



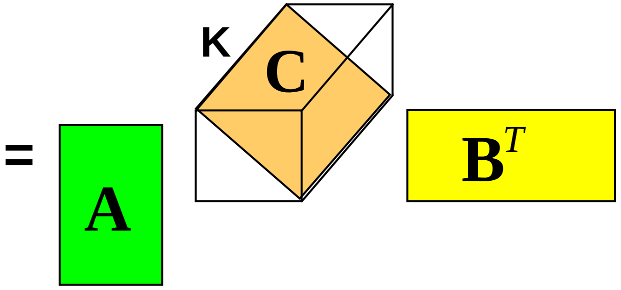
$\mathcal{H}$ is diagonal

(if i=j=k, h_{ijk}=1, else, h_{ijk}=0)



Sum of R rank-1 tensors:

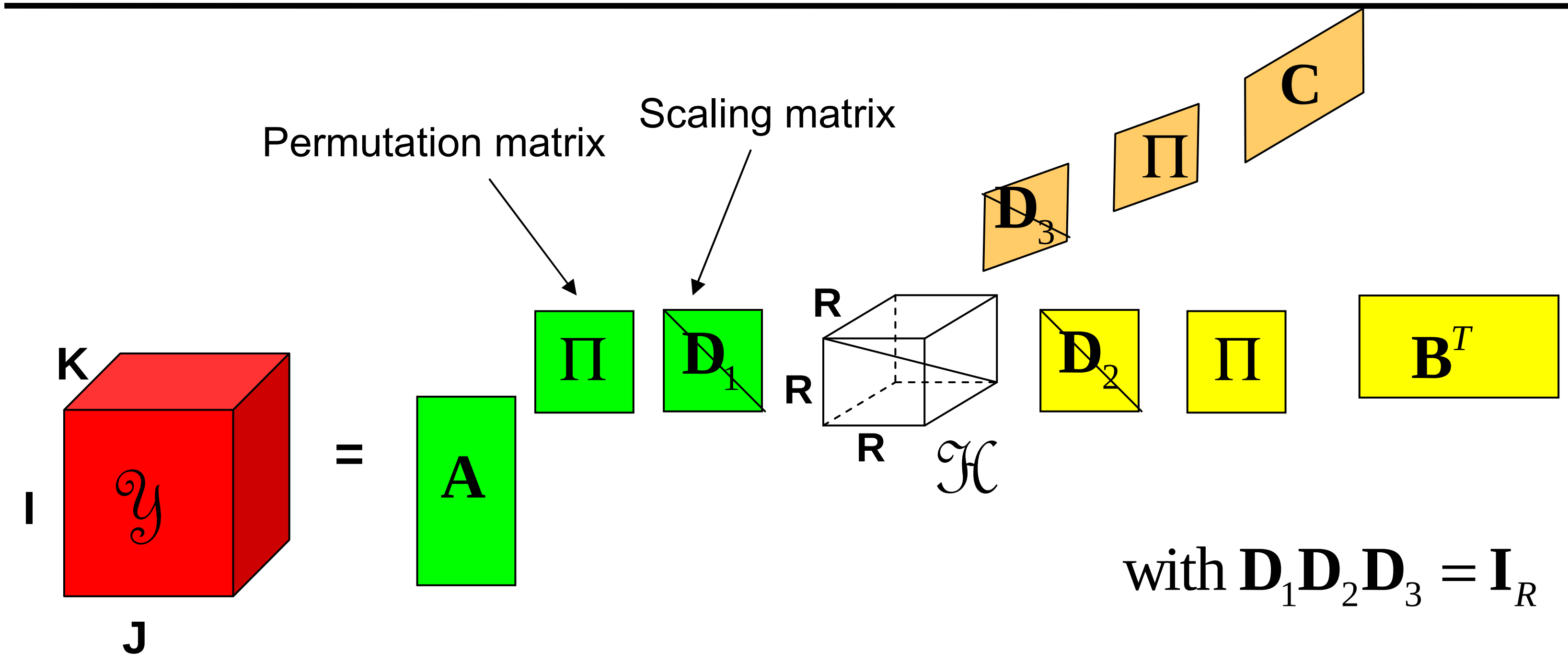
$\mathcal{Y}_1+\ldots+\mathcal{Y}_R$



$\mathcal{Y}$ = set of K matrices of the form:

$\mathcal{Y}(:,:,k)$=A diag(C(k,:)) B^T

Uniqueness of PARAFAC Decomposition (1)



with $\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3 = \mathbf{I}_R$

- Under mild conditons (next slide) PARAFAC is unique: only trivial ambiguities remain on **A**, **B** and **C** (permutation and scaling of columns).

- PARAFAC decomposition gives the **true** matrices **A**, **B** and **C** (up to the trivial ambiguities) → this is a key feature compared to matrix SVD (which gives only subspaces)

Uniqueness of PARAFAC Decomposition (2)

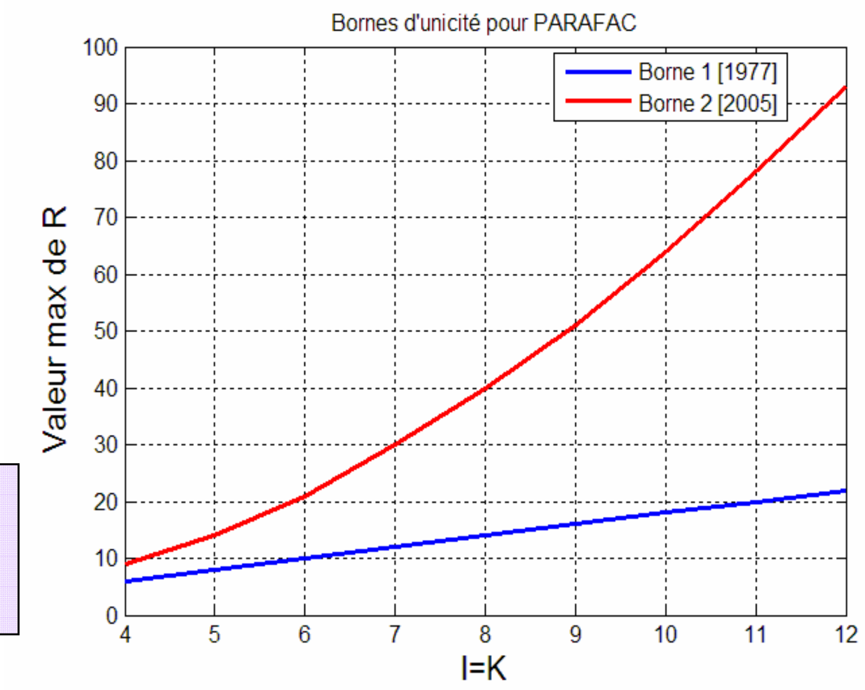
Uniqueness condition *[Kruskal, 1977]*

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2 \qquad (1)$$

k$_A$ is the Kruskal-rank of **A**

Generically, k_A=min(I,R)

$$\min(I,R)+\min(J,R)+\min(K,R)\geq 2(R+1) \qquad (2)$$

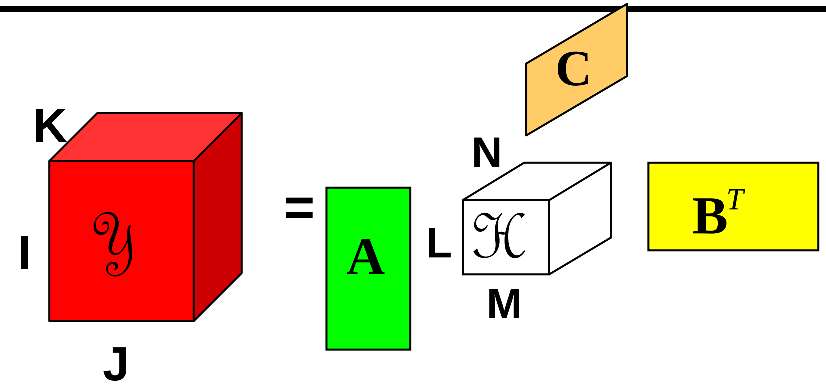


Rela e o n (real an comple cases) o n ( ) s o n ( )

[De Lathauwer 2005] :

$$J \geq R \quad \text{et} \quad \frac{I(I-1)}{2}\frac{K(K-1)}{2} \geq \frac{R(R-1)}{2} \qquad (3)$$

PARAFAC vs Tucker 3



PARAFAC

$$y_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$$

$\mathcal{H}$ is diagonal

L=M=N → A, B and C have the same nb. of columns

Unique (trivial ambiguities):
Only arbitrary scaling and permutation remains .

TUCKER 3

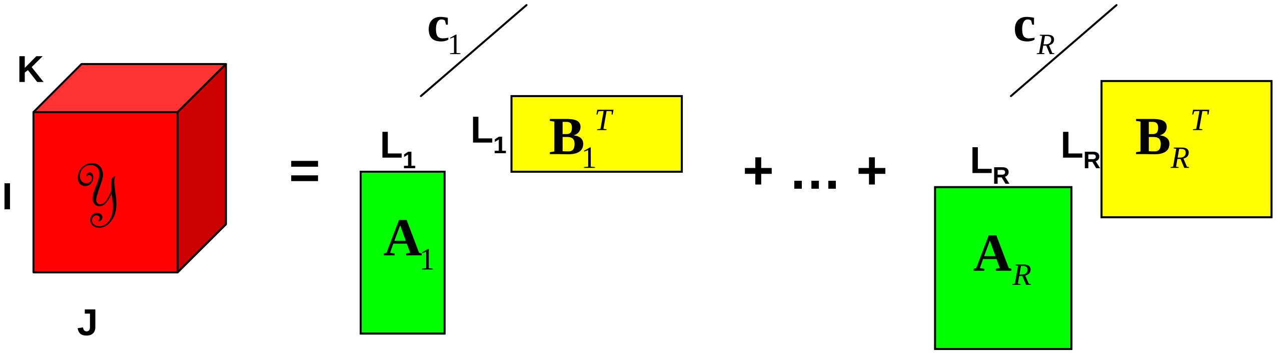
$$y_{ijk} = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} a_{il} b_{jm} c_{kn} h_{lmn}$$

$\mathcal{H}$ is not diagonal

$L \neq M \neq N$ → A, B and C do not necessarily have the same nb. of columns

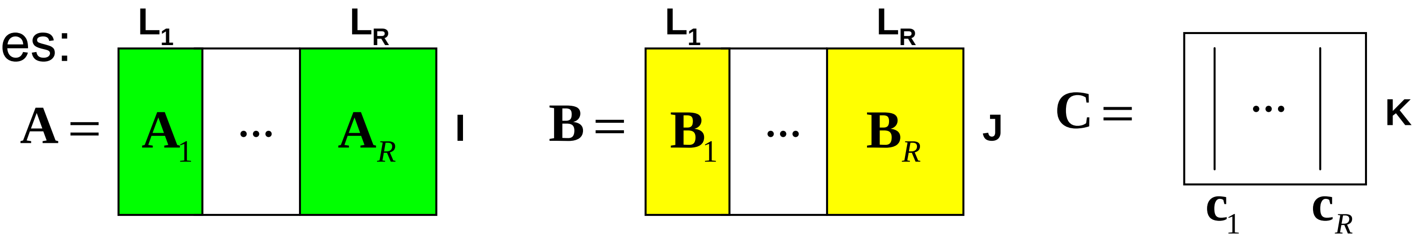
Not unique:
Rotational freedom still remains.

Block Component Decomposition in rank-(L_r,L_r,1) terms



BCD-(L_r,L_r,1)

- First generalization of PARAFAC in block terms [De Lathauwer, de Baynast, 2003] → If L_r=1 for all r, then BCD-(L_r,L_r,1)=PARAFAC
- Unknown matrices:



- BCD-(L_r,L_r,1) is said unique if the only remaining ambiguities are:

→ Arbitrary permutation of the blocks in A and B and of the columns of C

→ Rotational freedom of each block (block-wise subspace estimation) + scaling ambiguity on the columns of C

Uniqueness of the BCD-(L,L,1) (i.e., $L_1=L_2=\ldots=L_R=L$)

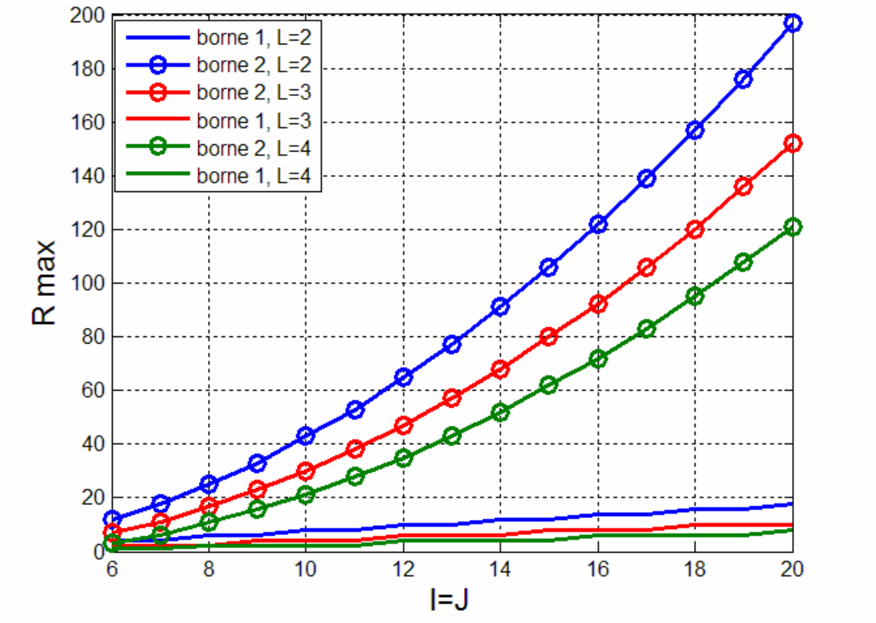
Sufficient bound 1 *[De Lathauwer SIMAX 2008]*

$$LR \leq IJ \text{ and } \min\left(\left\lfloor \frac{I}{L} \right\rfloor, R\right) + \min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + \min(K,R) \geq 2(R+1) \qquad (1)$$

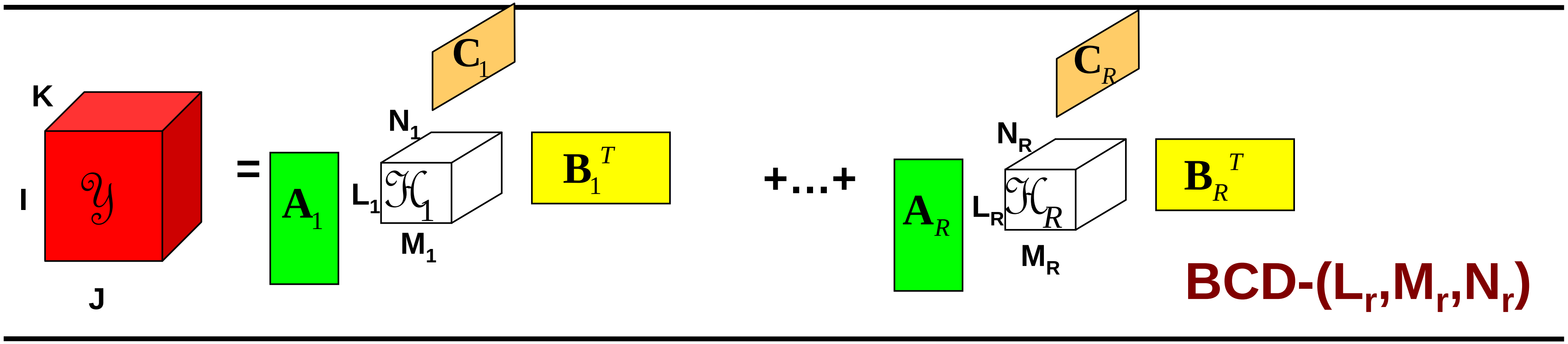
Sufficient bound 2 *[Nion, PhD Thesis, 2007]*:

$$R \leq \min(IJ, K) \text{ and } C_I^{L+1} . C_J^{L+1} \geq C_{R+L}^{L+1} - R \qquad (2)$$

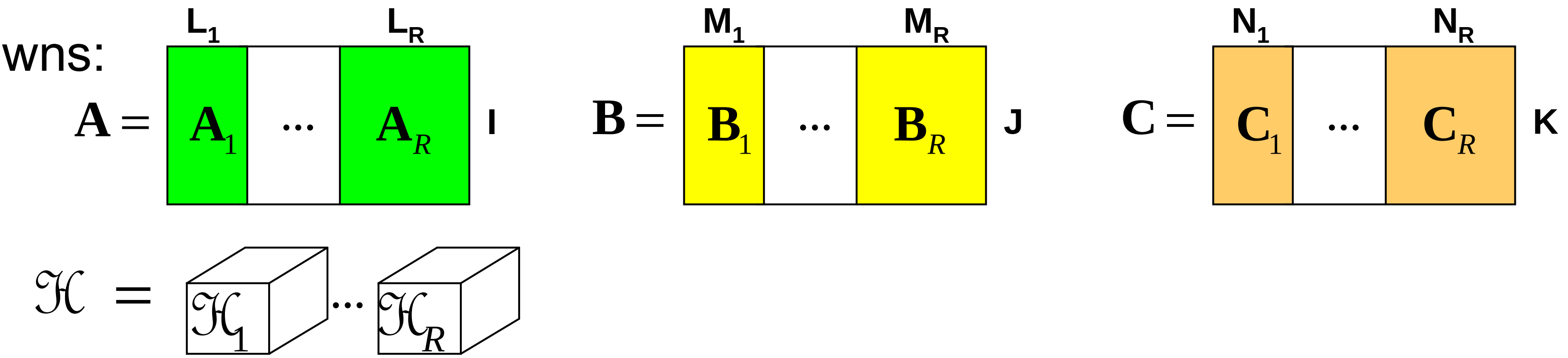
where $$C_n^k = \frac{n!}{k!(n-k)!}$$



Block Component Decomposition in rank-(L_r,M_r,N_r) terms



- Introduced by De Lathauwer in 2005
- Very General framework → generalization of PARAFAC, BCD-(L_r,L_r,1) and Tucker/HOSVD
- Sum of R Tucker decompositions
- Unknowns:



- Ambiguities: same as Tucker model for each of the R components

Roadmap

I. Introduction

II. A few Tensor Decompositions:
PARAFAC, HOSVD/Tucker, Block-Decompositions

III. Algorithms to compute Tensor Decompositions

IV. Applications

V. Conclusion and Future Research

Algorithms : basics

- Decompose $\mathcal{Y}$ ⟺ Estimate components **A**, **B** and **C**
- Minimization of the Frobenius norm of residuals

$$\Phi = \left\| \mathcal{Y} - Tens(\hat{\mathbf{H}}, \hat{\mathbf{S}}, \hat{\mathbf{A}}) \right\|_F^2$$

Tens = PARAFAC or BCD-(L,L,1) or BCD-(L,P,.)

Main idea: exploit the structure of the three matrix unfoldings simultanesouly

$$\left\{ \begin{array}{l} \mathbf{Y}_{\mathbf{K \times JI}} = \mathbf{C} \cdot \mathbf{Z}_1(\mathbf{B},\mathbf{A}) \\ \mathbf{Y}_{\mathbf{J \times IK}} = \mathbf{B} \cdot \mathbf{Z}_2(\mathbf{A},\mathbf{C}) \\ \mathbf{Y}_{\mathbf{I \times KJ}} = \mathbf{A} \cdot \mathbf{Z}_3(\mathbf{C},\mathbf{B}) \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \Phi = \left\| \mathbf{Y}_{\mathbf{K \times JI}} - \mathbf{C} \cdot \mathbf{Z}_1(\mathbf{B},\mathbf{A}) \right\|_F^2 \\ \Phi = \left\| \mathbf{Y}_{\mathbf{J \times IK}} - \mathbf{B} \cdot \mathbf{Z}_2(\mathbf{A},\mathbf{C}) \right\|_F^2 \\ \Phi = \left\| \mathbf{Y}_{\mathbf{I \times KJ}} - \mathbf{A} \cdot \mathbf{Z}_3(\mathbf{C},\mathbf{B}) \right\|_F^2 \end{array} \right.$$

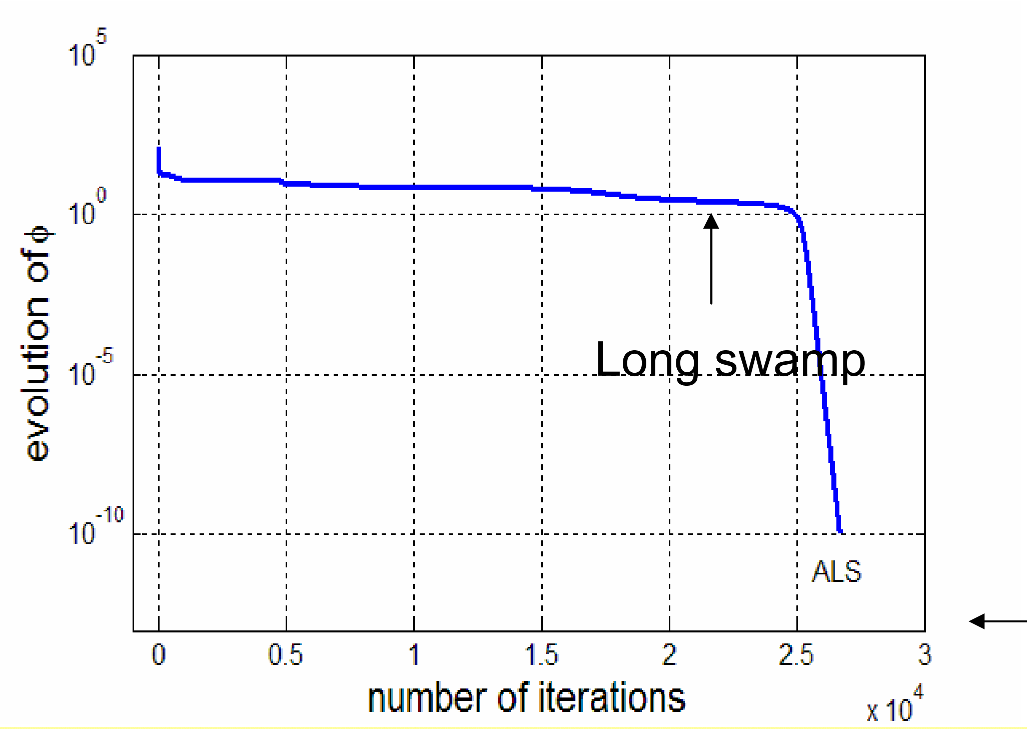
Z$_1$, **Z**$_2$ and **Z**$_3$ are built from 2 matrices only and their structure depends on the decomposition (PARAFAC, BCD-(L,L,1), etc)

ALS « Alternating Least Squares » algorithm

- ➢ Principle: Alternate updates of A=[A_1,…,A_R], B=[B_1,…,B_R] and C=[C_1,…,C_R] in the Least Squares sense.
- ➢ Each update = minimization of the cost function w.r.t. one the 3 matrix unfoldings

```
Initialisation: $\hat{\mathbf{A}}^{(0)}, \hat{\mathbf{B}}^{(0)}, k = 1$
while $\left|\Phi^{(k-1)} - \Phi^{(k)}\right| > \varepsilon$   (e.g. $\varepsilon = 10^{-6}$)
$\hat{\mathbf{C}}^{(k)} = \mathbf{Y}_{\mathbf{K \times JI}} \cdot \left[\mathbf{Z}_1(\hat{\mathbf{B}}^{(k-1)}, \hat{\mathbf{A}}^{(k-1)})\right]^{\dagger}$      (1)
$\hat{\mathbf{B}}^{(k)} = \mathbf{Y}_{\mathbf{J \times IK}} \cdot \left[\mathbf{Z}_2(\hat{\mathbf{A}}^{(k-1)}, \hat{\mathbf{C}}^{(k)})\right]^{\dagger}$      (2)
$\hat{\mathbf{A}}^{(k)} = \mathbf{Y}_{\mathbf{I \times KJ}} \cdot \left[\mathbf{Z}_3(\hat{\mathbf{C}}^{(k)}, \hat{\mathbf{B}}^{(k)})\right]^{\dagger}$      (3)
$k \leftarrow k + 1$
```

ALS algorithm: problem of swamps



Observation:

ALS is fast in many problems, but sometimes, a long swamp is encountered before convergence.

← 27000 iterations !

Long Swamps typically occur when:

- The loading matrices of the decomposition (i.e. the objective matrices) are ill-conditioned

-The updated matrices become ill-conditionned (impact of initialization)

- One of the R tensor-components in $\mathcal{Y} = \mathcal{Y}_1 + \ldots + \mathcal{Y}_R$ has a much higher norm than the R-1 others (e.g. « near-far » effect in telecommunications)

Improvement 1 of ALS: Line Search

Purpose: reduce the length of swamps

Principle: for each iteration, interpolate A, B and C from their estimates of 2 previous iterations and use the interpolated matrices in input of

1.Line Search:

$$\mathbf{C}^{(new)} = \mathbf{C}^{(k-2)} + \rho\,(\mathbf{C}^{(k-1)} - \mathbf{C}^{(k-2)})$$

$$\mathbf{B}^{(new)} = \mathbf{B}^{(k-2)} + \rho\,(\mathbf{B}^{(k-1)} - \mathbf{B}^{(k-2)})$$

$$\mathbf{A}^{(new)} = \mathbf{A}^{(k-2)} + \rho\,(\mathbf{A}^{(k-1)} - \mathbf{A}^{(k-2)})$$

Search directions

Choice of ρ crucial

ρ =1 annihilates LS step (i.e. we get standard ALS)

2.Then ALS update

$$\hat{\mathbf{C}}^{(k)} = \mathbf{Y}_{\mathbf{K\times JI}} \cdot \left[\mathbf{Z}_1(\hat{\mathbf{B}}^{(new)}, \hat{\mathbf{A}}^{(new)})\right]^{\dagger} \qquad (1)$$

$$\hat{\mathbf{B}}^{(k)} = \mathbf{Y}_{\mathbf{J\times IK}} \cdot \left[\mathbf{Z}_2(\hat{\mathbf{A}}^{(new)}, \hat{\mathbf{C}}^{(k)})\right]^{\dagger} \qquad (2)$$

$$\hat{\mathbf{A}}^{(k)} = \mathbf{Y}_{\mathbf{I\times KJ}} \cdot \left[\mathbf{Z}_3(\hat{\mathbf{C}}^{(k)}, \hat{\mathbf{B}}^{(k)})\right]^{\dagger} \qquad (3)$$

$$k \leftarrow k+1$$

Improvement 1 of ALS: Line Search

[Harshman, 1970] « LSH » Choose $\rho = 1.25$

[Bro, 1997] « LSB » Choose $\rho = k^{1/3}$ and validate LS step if decrease in Fit

[Rajih, Comon, 2005] « Enhanced Line Search (ELS) »

For REAL tensors $\Phi(\mathbf{A}^{(new)}, \mathbf{S}^{(new)}, \mathbf{H}^{(new)}) = \Phi(\rho) = 6^{th}$ order polynomial .

Optimal ρ is the root that minimizes $\Phi(\mathbf{A}^{(new)}, \mathbf{S}^{(new)}, \mathbf{H}^{(new)})$

[Nion, De Lathauwer, 2006]

«Enhanced Line Search with Complex Step (ELSCS) »

For complex tensors, look for optimal $\rho = m.e^{i\theta}$

We have $\Phi(\mathbf{A}^{(new)}, \mathbf{S}^{(new)}, \mathbf{H}^{(new)}) = \Phi(m, \theta)$

Alternate update of m and θ :

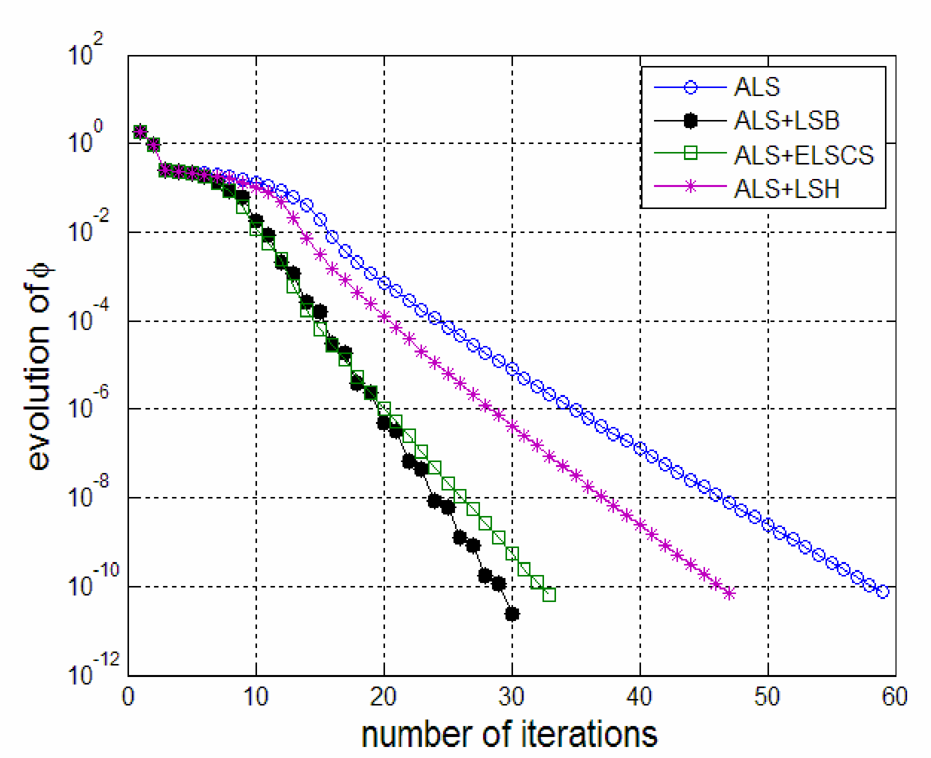
Update m : for θ fixed, $\dfrac{\partial \Phi(m,\theta)}{\partial m} = 5^{\text{th}}$ order polynomial in m

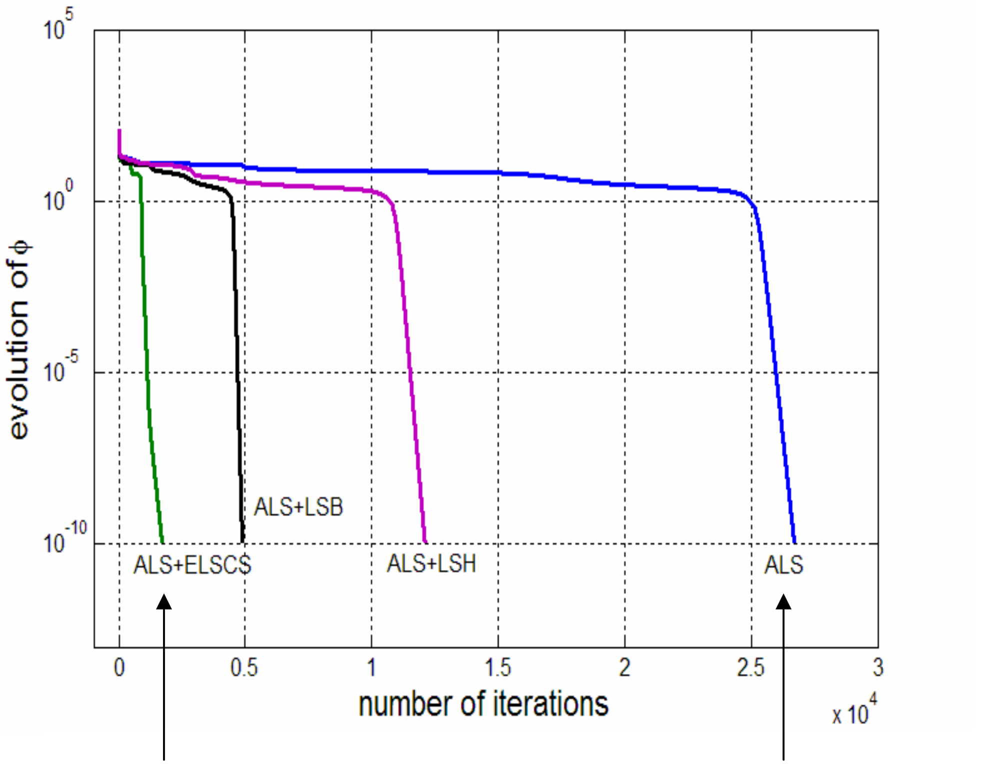
Update θ : for m fixed, $\dfrac{\partial \Phi(m,\theta)}{\partial \theta} = 6^{\text{th}}$ order polynomial in $t = \tan(\dfrac{\theta}{2})$

Improvement 1 of ALS: Line Search

«easy» problem

«difficult» problem



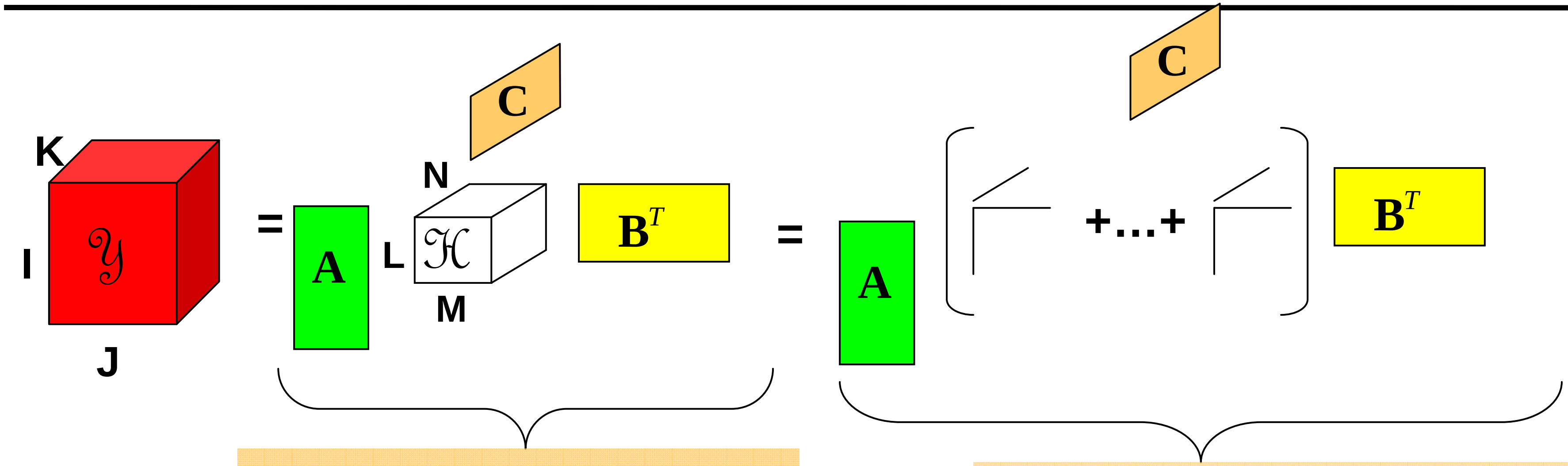


2000 iterations

27000 iterations

➢ Line Search → Large reduction of the number of iterations at a very low additional complexity w.r.t. standard ALS

Improvement 2 of ALS: Compression



STEP 1:

Fit a Tucker Model on $\mathcal{Y}$

STEP 2:

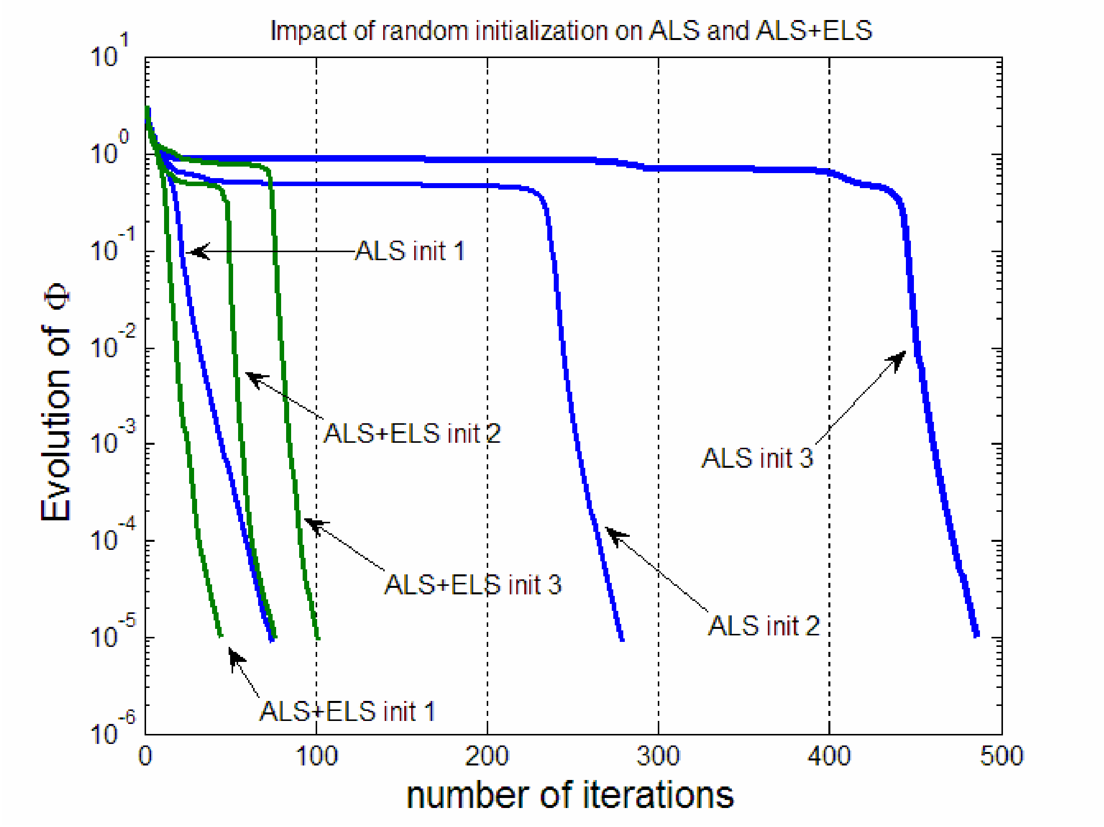
Fit the model on the small core tensor $\mathcal{H}$ (compressed space)

STEP 3:

Come back to original space

➢ Compression → Large reduction of the cost per iteration since the model is fitted in compressed space.

Improvement 3 of ALS: Good initialization



Comparison ALS and ALS+ELS, with three random initializations

Instead of using random initializations, could we use the observed tensor itself ?

Improvement 3 of ALS: Good initialization

Slices $\mathbf{Y}_k$ (IxJ) of $\mathcal{Y}$:
$$\begin{cases} \mathbf{Y}_1 = \mathbf{H} \cdot \Lambda_1 \cdot \mathbf{S}^T \\ \mathbf{Y}_2 = \mathbf{H} \cdot \Lambda_2 \cdot \mathbf{S}^T \\ \vdots \qquad \vdots \\ \mathbf{Y}_K = \mathbf{H} \cdot \Lambda_K \cdot \mathbf{S}^T \end{cases}$$
, where the Λ_i are diagonal

For PARAFAC: if $R \leq \min(I, J)$, the slices $\mathbf{Y}_k$ are generically rank-R

For any pair (k$_1$, k$_2$) : $\mathbf{Y}_{k_1} \cdot (\mathbf{Y}_{k_2})^\dagger = \mathbf{H} \cdot (\Lambda_{k_1} \cdot \Lambda_{k_2}{}^{-1}) \cdot \mathbf{H}^\dagger$

Estimate $\hat{\mathbf{H}}^{(0)}$ as the R principal eigenvectors. Then deduce $\hat{\mathbf{S}}^{(0)}$ and $\hat{\mathbf{A}}^{(0)}$

→ Called Direct Trilinear Decomposition (DTLD)

→ If no noise, the model is exact DTLD gives the exact solution.

→ If noise is present, DTLD gives a good initialization

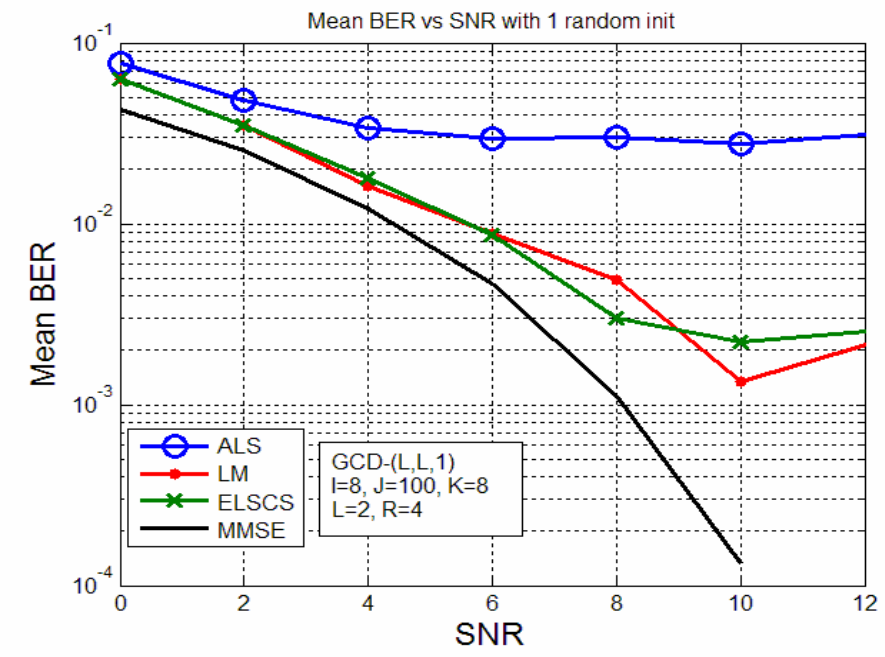
→ The same holds for Block Component Decompositions
(via generalization of DTLD)

→ To keep in mind: can only be used if at least 2 dimensions are long enough (For PARAFAC: $R \leq \min(I, J)$)

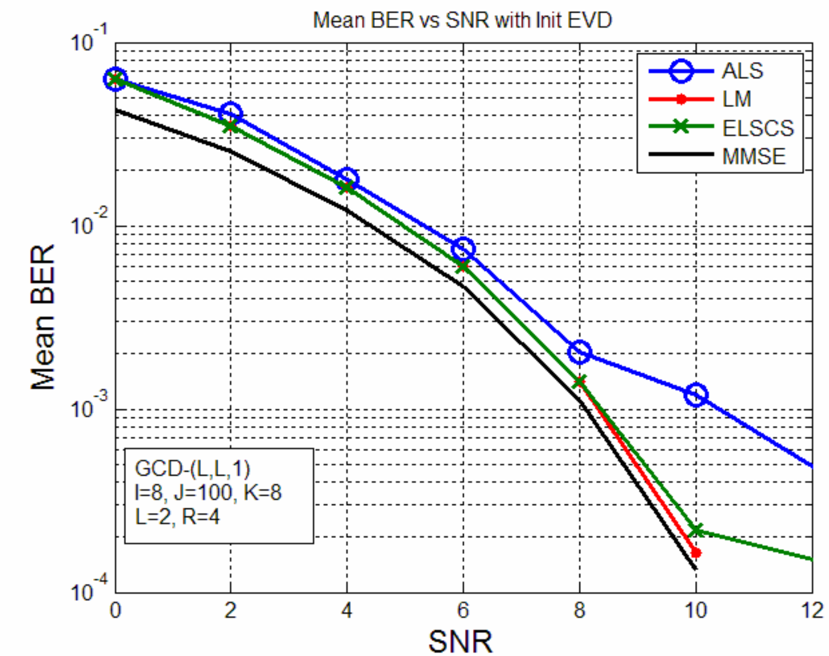
Improvement 3 of ALS: Good initialization

Simulations with BCD-(L,L,1), I=8, J=100, K=8, L=2, R=4

One random initialization



One initialization via DTLD



→ If dimensions allow it, use the DTLD-initialization + only 2 or 3 random initializations

→ Else, use e.g., 10 random initializations

→ It does not make sense to draw general conclusions on the average performance (e.g. BER curves with Monte Carlo runs) with only one initialization.

Concluding remarks on algorithms

→ Standard ALS sometimes slow (swamps)

→ ALS+ELS (sometimes drastically) reduces swamp length at low additional complexity

→ Other algorithms: e.g. Levenberg-Marquardt → convergence very fast, not very sensitive to ill-conditioned data, but higher complexity and memory (dimensions of Jacobian matrix=IJK)

→ <u>Important practical considerations:</u>

- Dimensionality reduction pre-processing step (via Tucker/HOSVD)
- Initialization via DTLD if possible

→ Algorithms have to be adapted to include constraints specific to applications:

- preservation of specific matrix-structures (Toeplitz, Van der Monde, etc)
- Constant Modulus, Finite Alphabet, …
- non-negativity constraints (e.g. Chemometrics applications)

Roadmap

I. Introduction

II. A few Tensor Decompositions:
PARAFAC, HOSVD/Tucker, Block-Decompositions

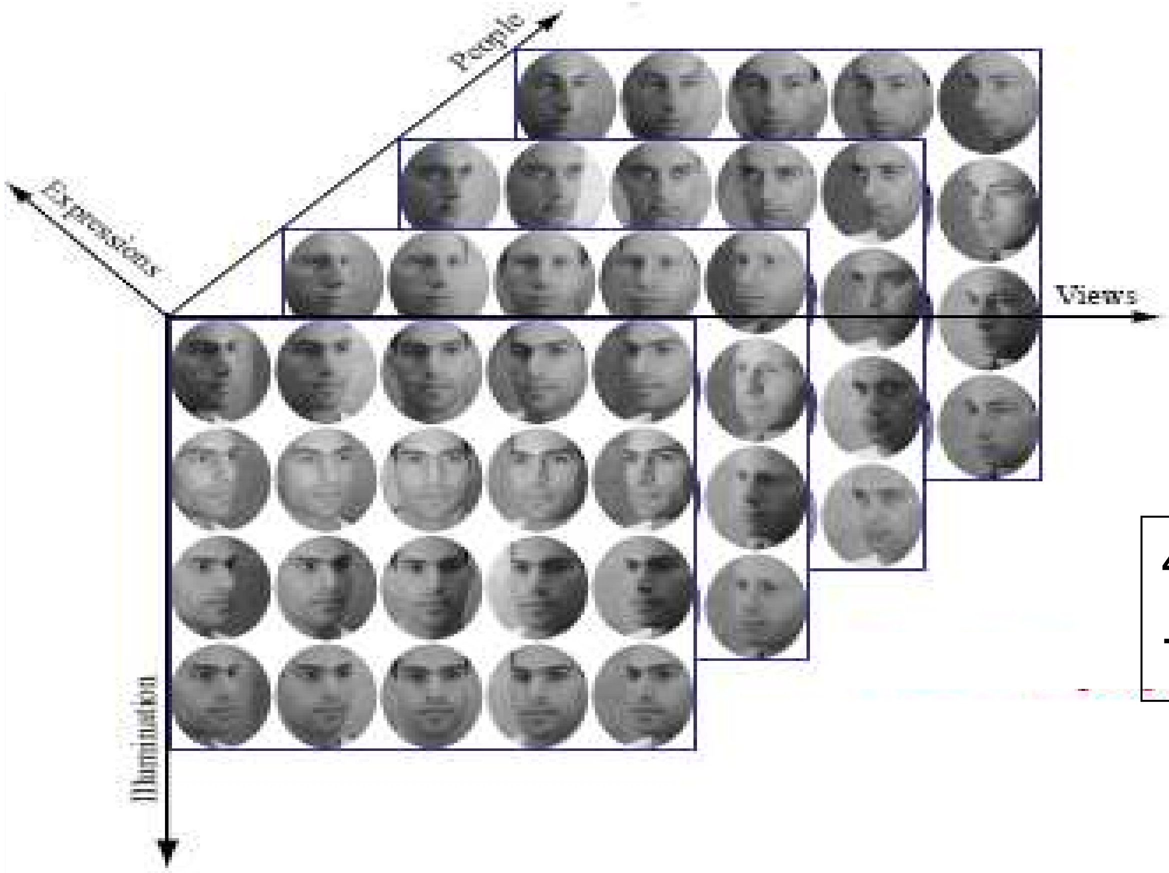
III. Algorithms to compute Tensor Decompositions

IV. Applications

V. Conclusion and Future Research

Application 1: Tensor Faces & Face Recognition

[Vasilescu & Terzopoulos, 2003]



Learning Database:

28 People
3 Expressions
5 Viewpoints
3 Illuminations

↓

45 images per person

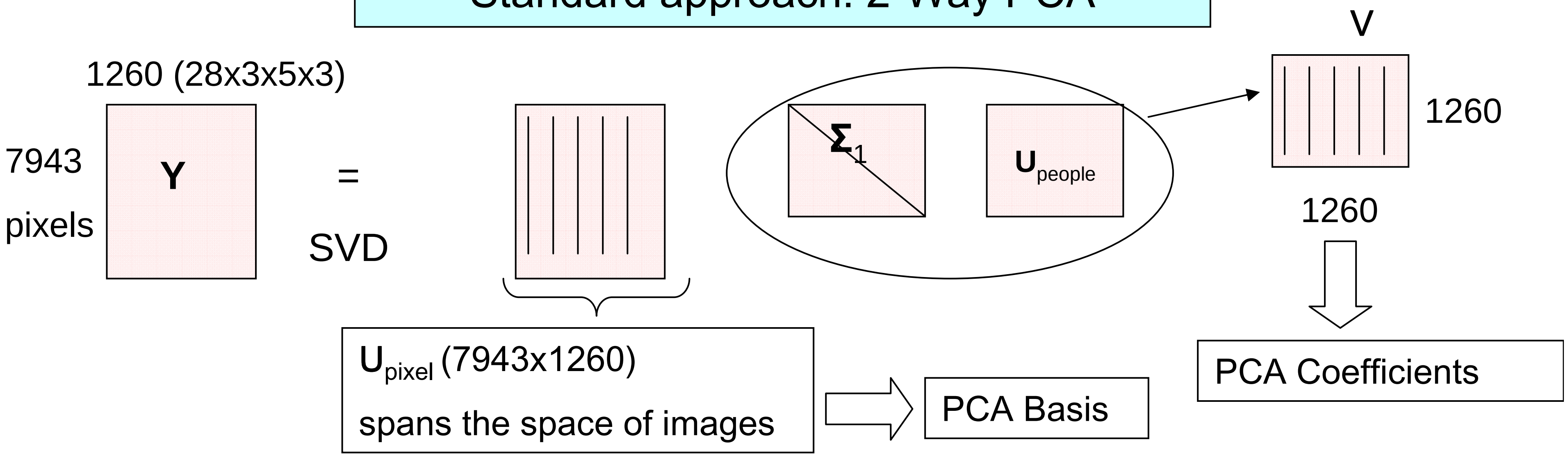
7943 pixels per image

Objective: associate input image (7943x1) to one of the 28 people

Application 1: Tensor Faces & Face Recognition

[Vasilescu & Terzopoulos, 2003]

Standard approach: 2-Way PCA



U_{pixel} (7943x1260)

spans the space of images

PCA Basis

PCA Coefficients

→ 1 image represented by one vector of 1260 coefficients in V

→ 1 person represented by a set of 45 vectors in V

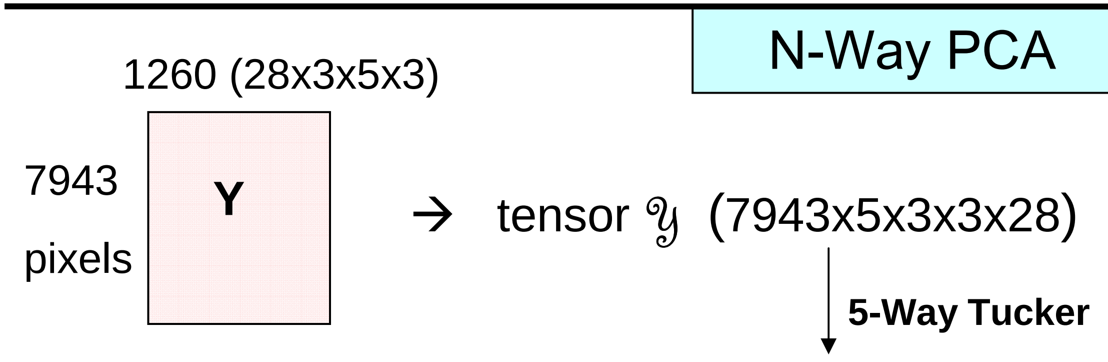
Input Image **d** (7943x1)

1) Projection of **d** in the space of PCA coefficients: $\mathbf{c} = U^H_{pixel}\mathbf{d}$ (1260x1)

2) $\min_i ||\mathbf{c} - \mathbf{v}_i||$ to associate score vector **c** to one person

Application 1: Tensor Faces & Face Recognition [Vasilescu & Terzopoulos, 2003]

N-Way PCA



$$\mathcal{Y} = \mathcal{H} \times_1 \boldsymbol{U}_{pixels} \times_2 \boldsymbol{U}_{views} \times_3 \boldsymbol{U}_{illums} \times_4 \boldsymbol{U}_{express} \times_5 \boldsymbol{U}_{people}$$

U_{pixels} (7943x7943) spans the space of images

U_{views} (5x5) spans the space of viewpoint parameters

U_{illums} (3x3) spans the space of illumination parameters

$U_{express}$ (3x3) spans the space of expression parameters

U_{people} (28x28) spans the space of people parameters

→ $\mathcal{H}$ describes how the different modes interact

→Compression flexibility: greater control than 2-Way PCA (truncation of the different bases independently)

Application 1: Tensor Faces & Face Recognition

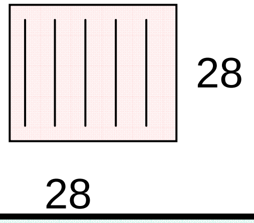
[Vasilescu & Terzopoulos, 2003]

N-Way PCA

$$\mathcal{Y} = \mathcal{H} \times_1 \boldsymbol{U}_{pixels} \times_2 \boldsymbol{U}_{views} \times_3 \boldsymbol{U}_{illums} \times_4 \boldsymbol{U}_{express} \times_5 \boldsymbol{U}_{people}$$

$$= \mathcal{B} \times_5 \boldsymbol{U}_{people}$$

7943x5x3x3x28



1) For all triplets (view,illums,express), build the basis $\mathbf{B}_{\mathbf{v,i,e}}$ (7943x28) and project unknown image

$$\mathbf{c} = \mathbf{B}_{v,i,e}^{\dagger}\, \mathbf{d}$$

2) Compare the 28x1 score vector **c** to the loadings in $\mathbf{U}_{people}$

$$\min_i \|\mathbf{c}-\mathbf{u}_i\|$$

to associate the input image **d** to one of the 28 persons

Performance comparison (recognition rate):

2-Way PCA   27%   5-Way PCA: 88%

Application 2: Chemometrics- Analysis of fluorescence data via PARAFAC [R. Bro, 1997]

Data set:

→2 chemical samples, each containing different and unknown concentrations of 3 unknown chemical components.

Goal:

→ Find which chemical components are present in the samples

Method: fluorescence

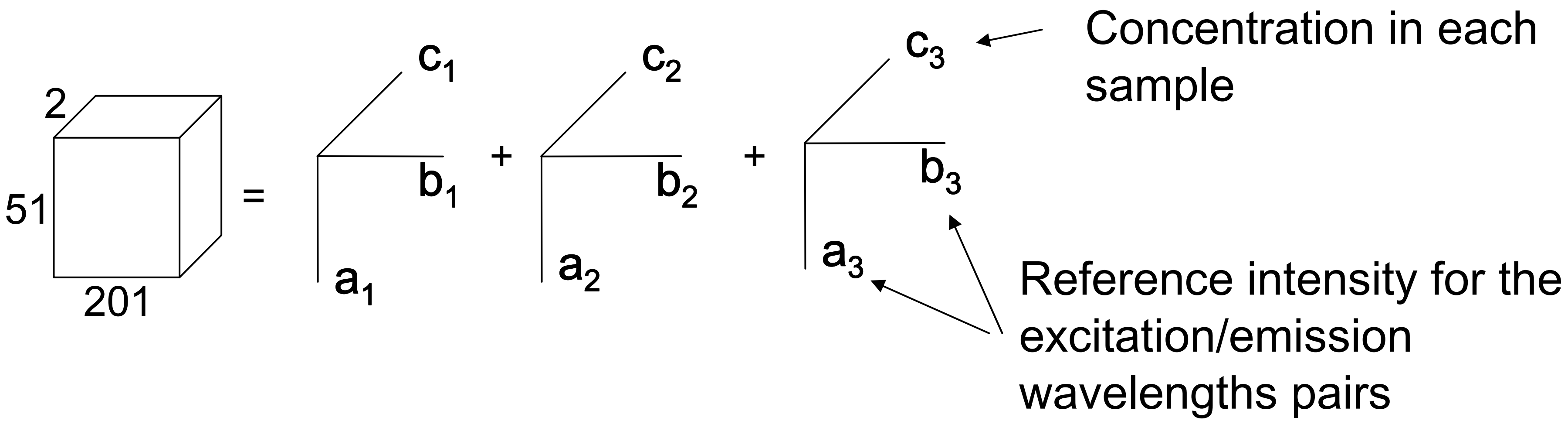
→Excitation of the samples with 51 wavelengths (250-300nm)

→Measure of the intensity of emission over 201 wavelengths (250-450nm)

Application 2: Chemometrics- Analysis of fluorescence data via PARAFAC [R. Bro, 1997]

Data cube $\mathcal{Y}$ (51x201x2): holds the whole set of measured intensities, for the two samples

Fit PARAFAC model with R=3 components

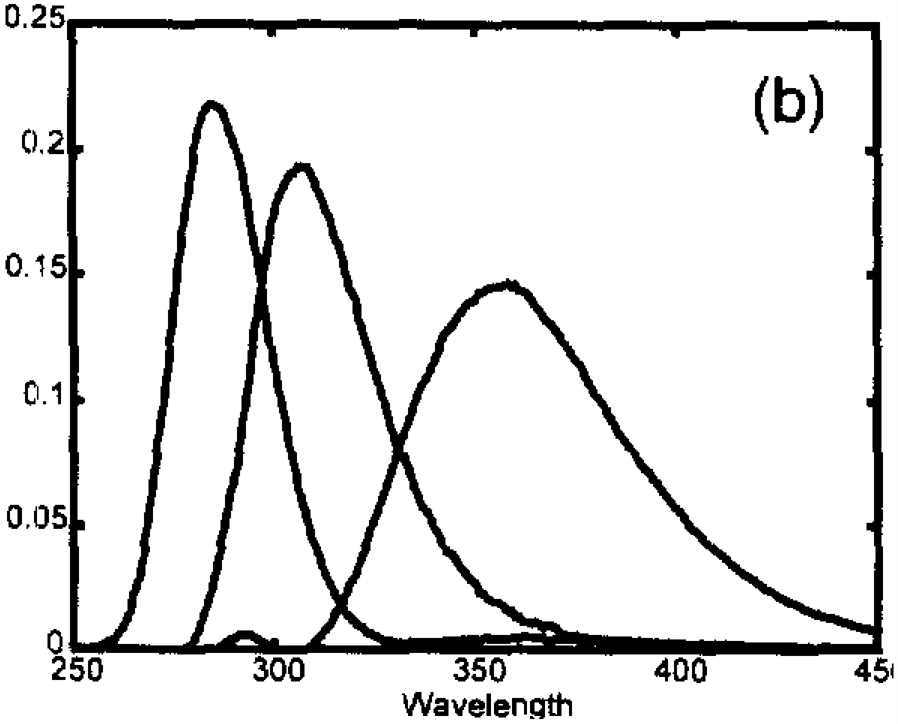

2
51
201
=
c_1
b_1
a_1
+
c_2
b_2
a_2
+
c_3
b_3
a_3
Concentration in each sample
Reference intensity for the excitation/emission wavelengths pairs

Identification of 3 chemical components with only 2 samples

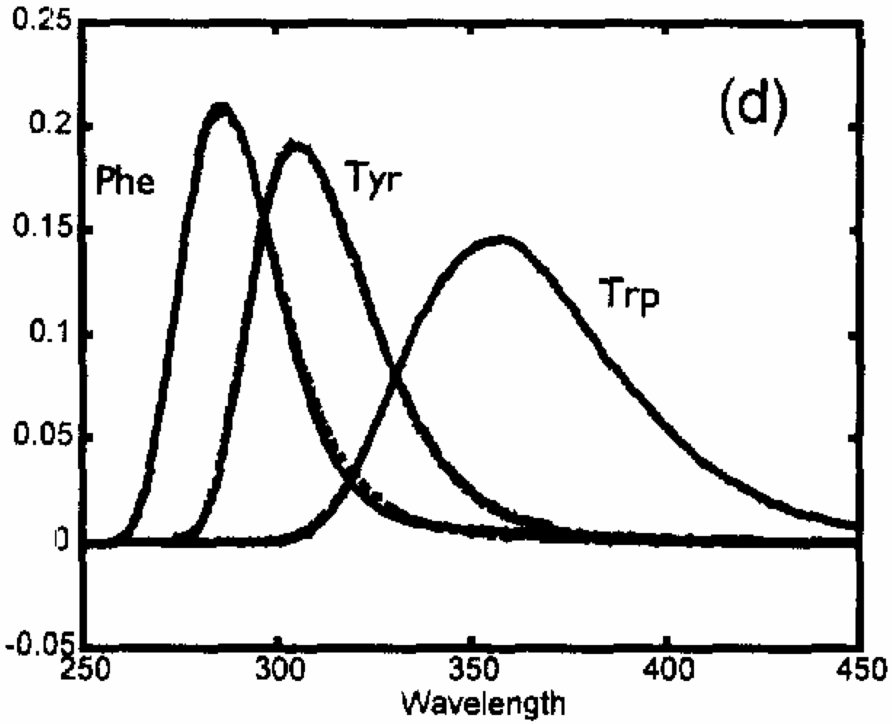
→ thanks to uniqueness of PARAFAC decomposition

Application 2: Chemometrics- Analysis of fluorescence data via PARAFAC [R. Bro, 1997]

Estimated emission spectrum



True excitation spectrum



Results from paper « PARAFAC: tutorial and applications », by Rasmus Bro, 1997

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization

CDMA (« Code Division Multiple Access »)

→ Used in 3rd generation standard (UMTS)

→ Allows users to communicate *simultaneously* in the *same bandwidth*

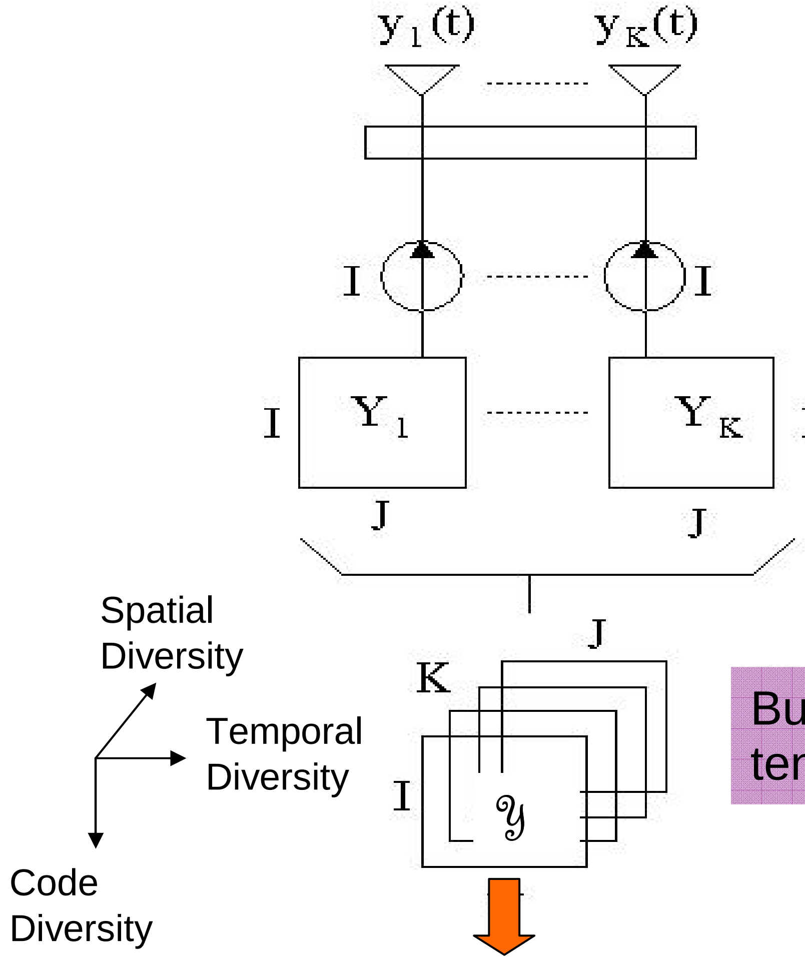
User 1 wants to transmit $s_1=[1\ \ -1\ \ -1]$.

→ CDMA code allocated to user 1: $c_1=[1\ \ -1\ \ \ 1\ \ -1]$.

→ User 1 transmits $[+c_1\ \ \ -c_1\ \ \ \ -c_1]$

→ User 2 transmits his symbols spread by his own CDMA code c_2 orthogonal to c_1, etc

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization



K receive antennas

Chip rate sampling (I times faster than symbol rate)

Observation during J symbol periods

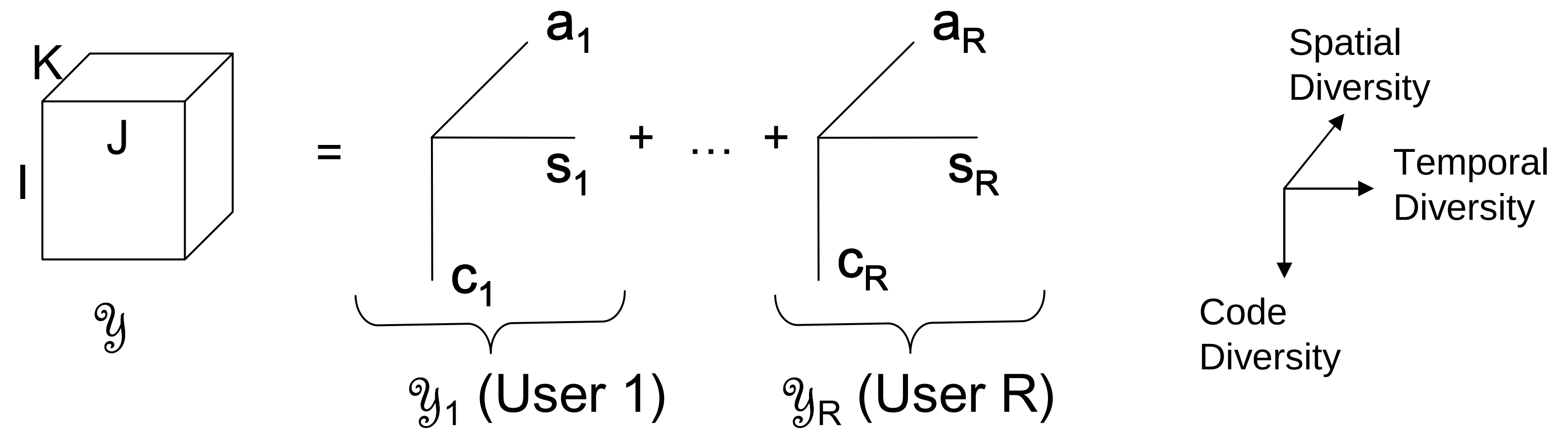
Build the 3rd order observed tensor $\mathcal{Y}$

Decompose $\mathcal{Y}$ to blindly estimate the transmitted symbols.
Which decomposition to use? → the one that best reflects the algebraic structure of the data

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization

Case 1: single path propagation (no inter-symbol-interference)

[Sidiropoulos et al., 2001]



I = length of the CDMA codes

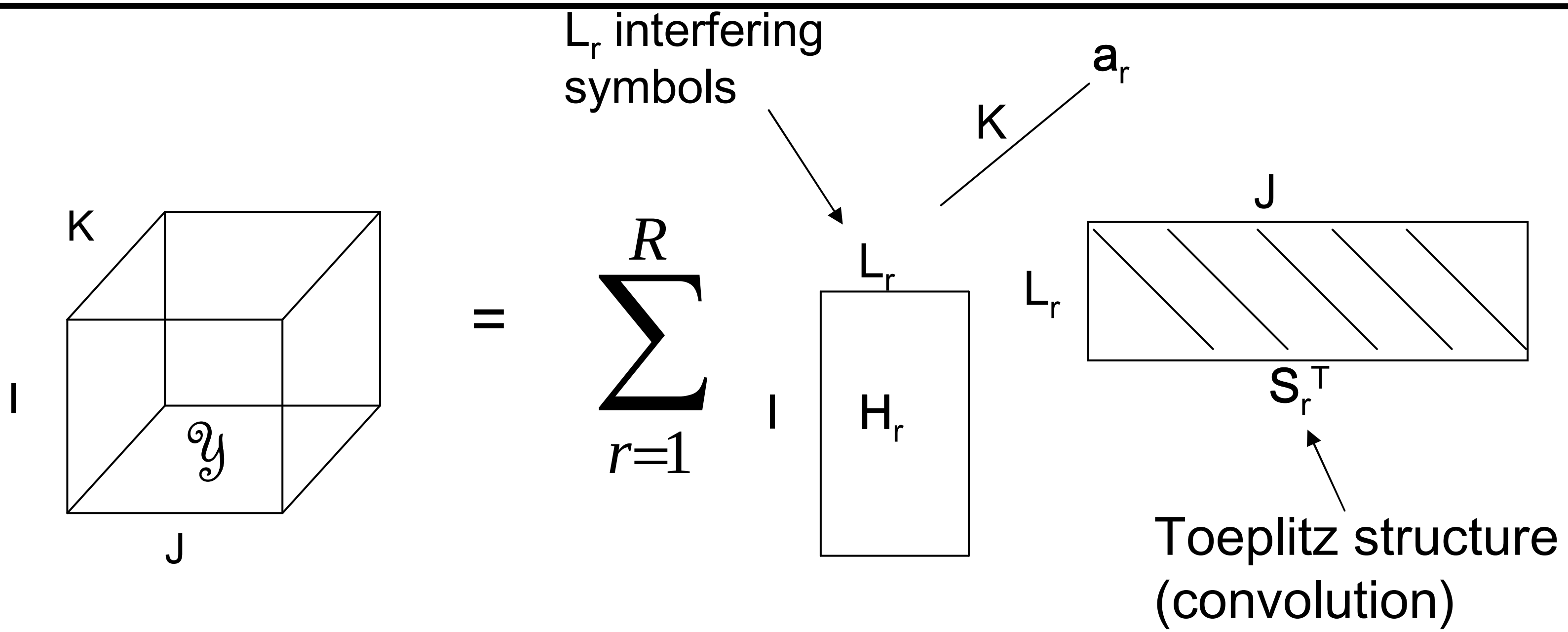
J = number of symbols

K = number of antennas at the receiver

« Blind » receiver: uniqueness of PARAFAC does not require prior knowledge of the CDMA codes, neither of pilot sequences to **blindly estimate the symbols of all users.**

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization

Case 2: Multi-path propagation with inter-symbol-interference but far-field reflections only [De Lathauwer & de Baynast 2003]



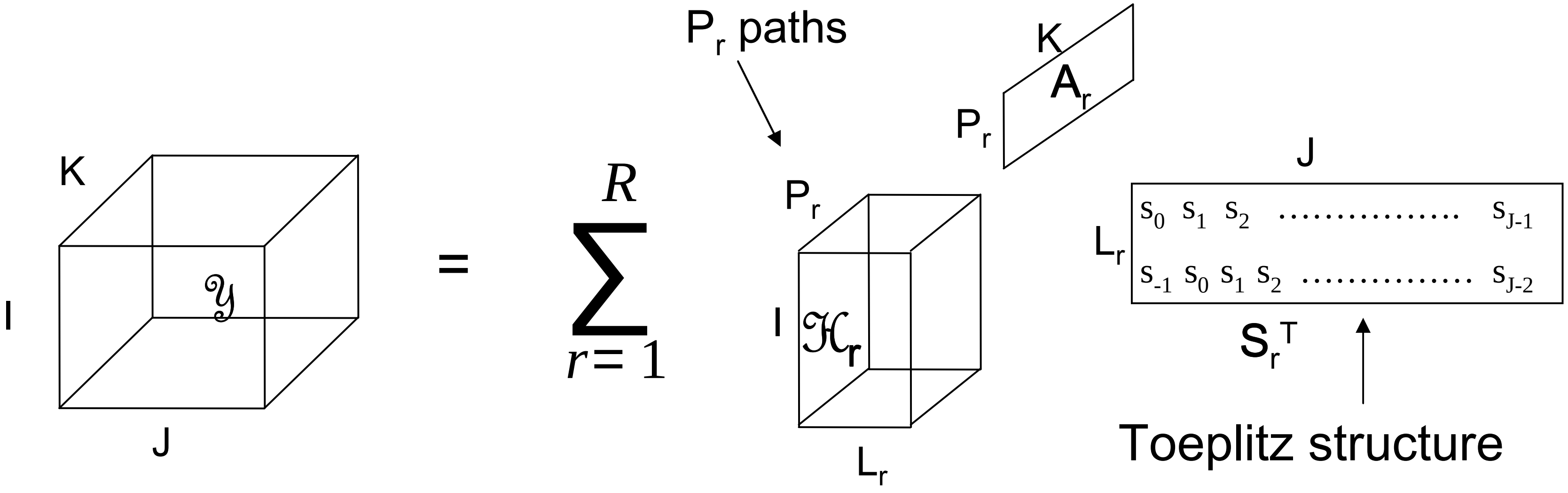
H_r → Channel matrix (channel impulse response convolved with CDMA code)

S_r → Symbol matrix, holds the J symbols of interest for user r

a_r → Response of the K antennas to the angle of arrival (steering vector)

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization

Case 3: Multi-path propagation with inter-symbol-interference but reflections not only in the far field [Nion & De Lathauwer 2006]



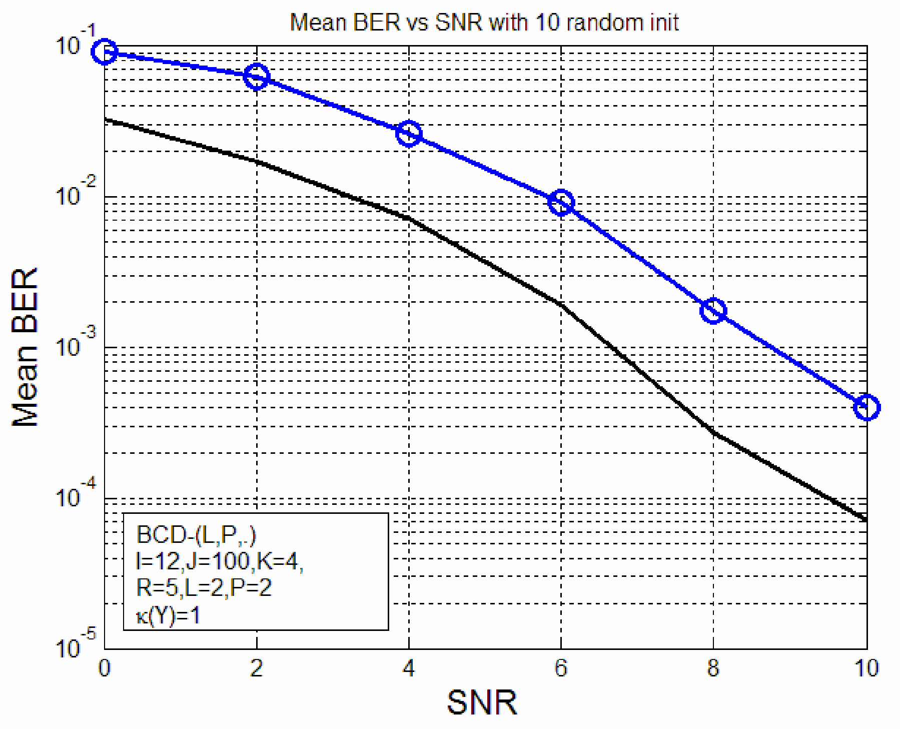
$\mathcal{H}_r$ → Channel matrix (channel impulse response convolved with CDMA code)

S_r → Symbol matrix, holds the J symbols of interest for user r

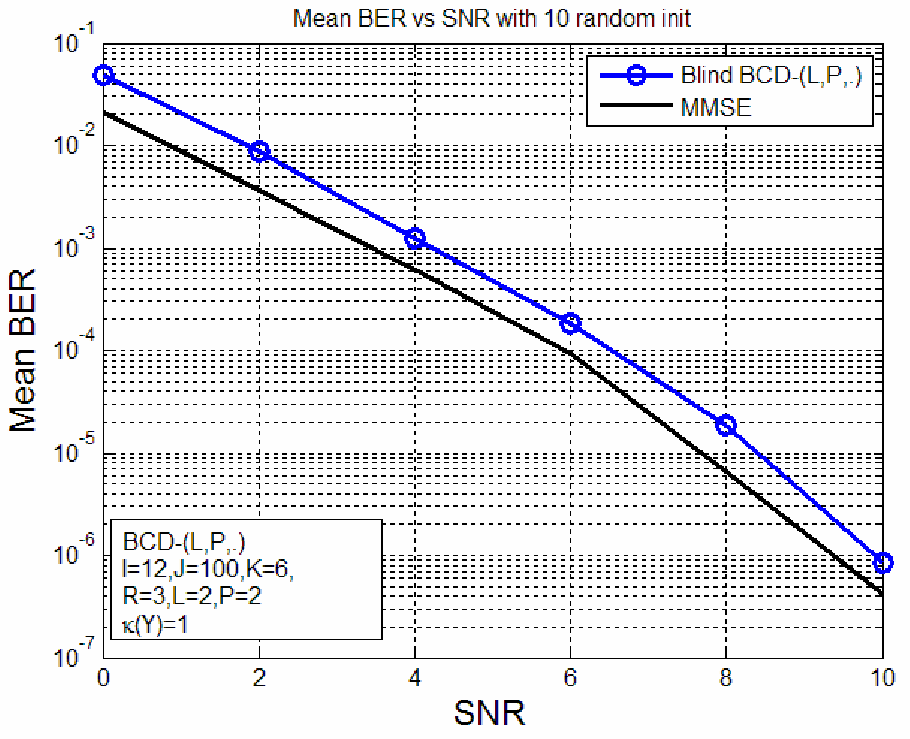
A_r → Response of the K antennas to the angles of arrival (steering vectors)

Application 3: Telecommunications - Blind CDMA system via PARAFAC and its generalization

BCD-(L,P,.) with I=12, J=100, L=2, P=2 and 10 random initializations.



K=4 antennas and R=5 users

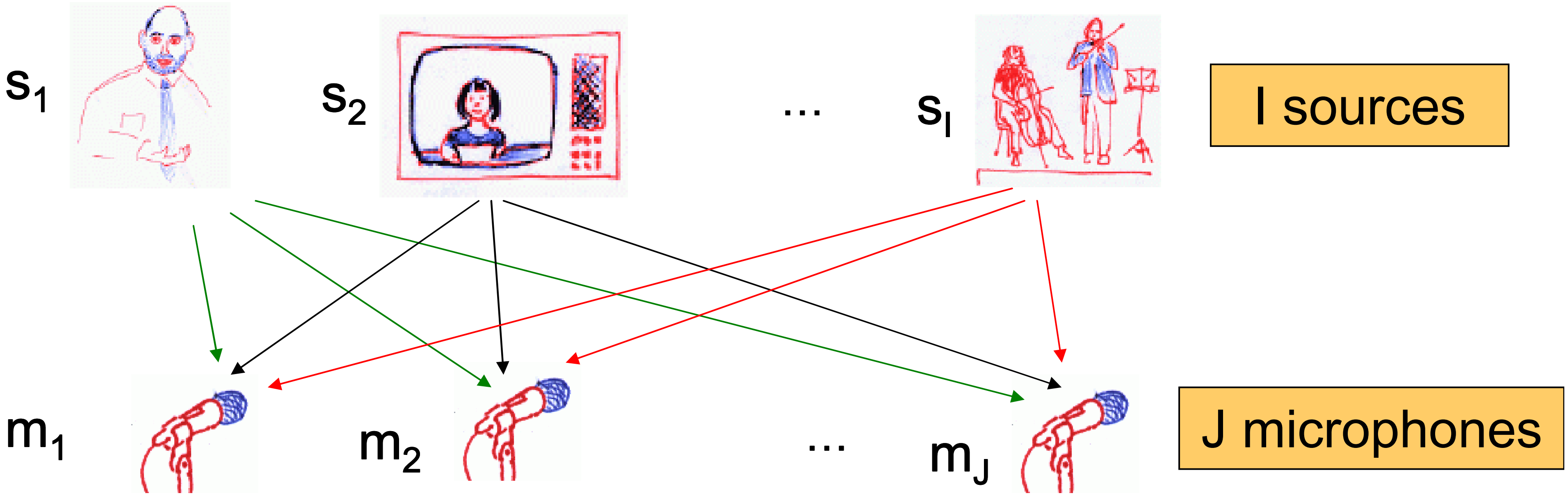


K=6 antennas and R=3 users

Application 4:

Blind Source Separation (instantaneous mixtures)

« Cocktail Party Problem »



s_1 s_2 … s_I

I sources

m_1 m_2 … m_J

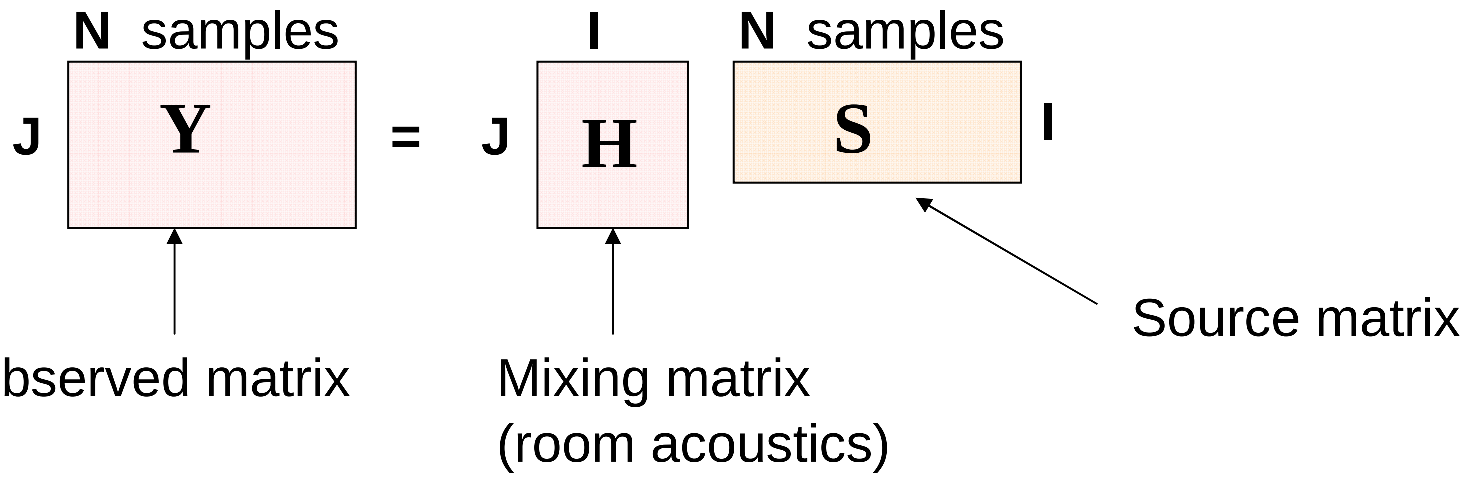
J microphones

Goal: estimate the I unknown sources $s_1, \dots, s_I$, from the J recordings $m_1, \dots, m_J$ *only* . (« blind source separation (BSS)»)

Application 4:

Blind Source Separation (instantaneous mixtures)

Data Model for linear instantaneous mixtures:



N samples

J $\mathbf{Y}$ **=** **J** $\mathbf{H}$

I

N samples

$\mathbf{S}$ **I**

Observed matrix

Mixing matrix
(room acoustics)

Source matrix

<u>Issues:</u>

→ How to find **H** and **S** ?

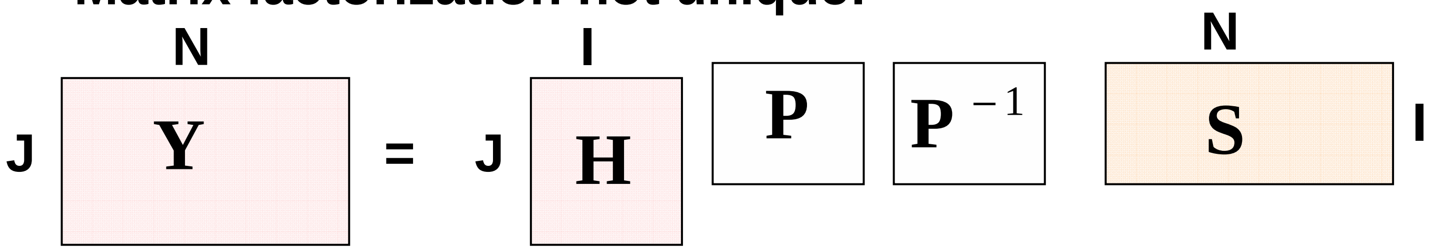
→ What happens if we have more sources than sensors (I>J)
(« under-determined case ») **H** is fat so not left-pseudo invertible.

→ What about convolutive mixtures (to take reverberations on walls into account)?

Application 4:

Blind Source Separation (instantaneous mixtures)

Matrix factorization not unique:



The SVD of **Y** would give us the subspaces that generate **H** and **S**, but not **H** and **S** themselves → **We need more assumptions**!

Assumption: The I sources are statistically independent

« Independent Component Analysis » (ICA), [Comon, 1994].

⇨ Find **H** that makes the source estimates as much independent as possible.

⇨ Use of Second-Order or Higher-Order Statistics (SOS or HOS)

+ Application-specific assumptions to reduce the ambiguity:

- Matrix-Structures (Toeplitz, Van Der Monde,…)
- Finite Alphabet (Symbol constellation), Constant Modulus, etc

Application 4:

Blind Source Separation (instantaneous mixtures)

« Second-Order-Blind-Identification » (SOBI) [Belouchrani et al. 1997]

$$
\begin{aligned}
\mathbf{C}_k &= E[\mathbf{y}_t \mathbf{y}_{t-\tau_k}^H] \\
&= \mathbf{H} E[\mathbf{s}_t \mathbf{s}_{t-\tau_k}^H] \mathbf{H}^H \\
&= \mathbf{H} \underset{\substack{\uparrow \\ \boxed{\textbf{diagonal}}}}{\mathbf{D}_k} \mathbf{H}^H
\end{aligned}
\quad \Longrightarrow \quad
\left\{
\begin{array}{l}
\mathbf{C}_1 = \mathbf{H}\mathbf{D}_1\mathbf{H}^H \\
\ \vdots \qquad\quad \vdots \\
\mathbf{C}_K = \mathbf{H}\mathbf{D}_K\mathbf{H}^H
\end{array}
\right.
$$

K delays → K covariance matrices

$$\Downarrow$$

Use existing algorithms for Joint Diagonalization of a set of matrices to find H

SOBI relies on simultaneous diagonalization algorithms → does not work in under-determined cases (i.e., when **H** is fat)

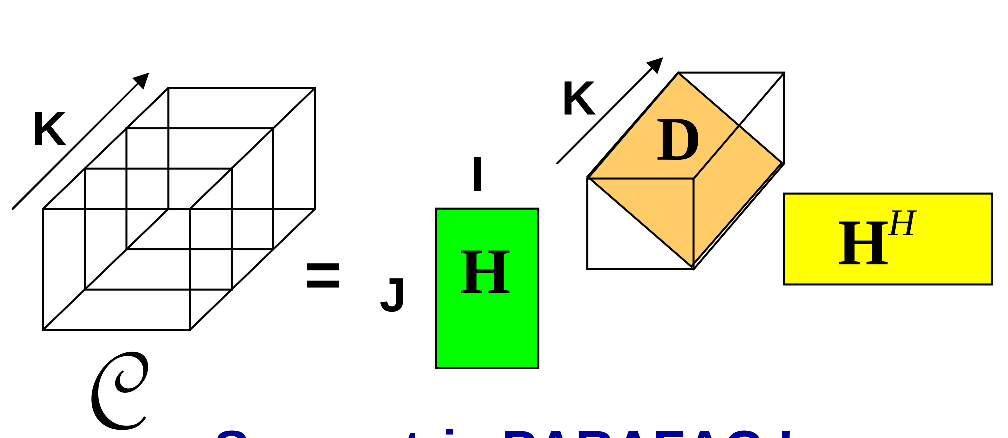
Application 4:

Blind Source Separation (instantaneous mixtures)

« Second-Order-Blind-Identification of Under-determined mixtures » (SOBIUM)

[Castaing & De Lathauwer 2006]

$$\begin{cases} \mathbf{C}_1 = \mathbf{H}\mathbf{D}_1\mathbf{H}^H \\ \vdots \qquad\qquad \vdots \\ \mathbf{C}_K = \mathbf{H}\mathbf{D}_K\mathbf{H}^H \end{cases} \Leftrightarrow$$



Symmetric PARAFAC !

→Lower complexity than SOBI: Tucker compression in mode 3 before fitting the PARAFAC model (K reduced to I) to find **H**

→ Works for under-determined cases (uniqueness of PARAFAC):

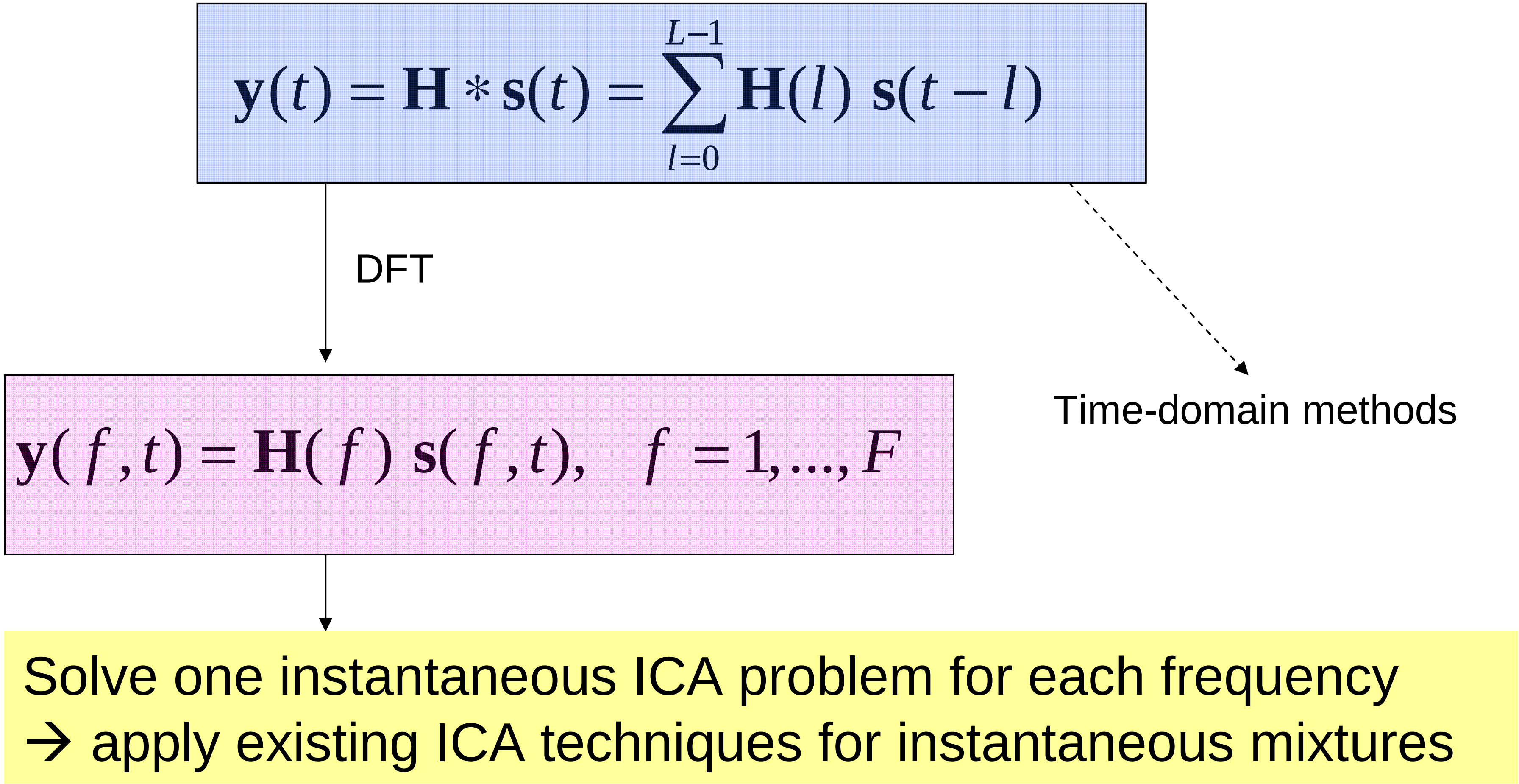
J	2	3	4	5	6	7	8
I_{max}	2	4	6	10	15	20	26

Application 5:

Blind Source Separation (convolutive mixtures)

Y=HS → instantaneous mixtures

Multiple reverberations on the walls → separation of convolutive mixture



$$\mathbf{y}(t) = \mathbf{H} * \mathbf{s}(t) = \sum_{l=0}^{L-1} \mathbf{H}(l)\ \mathbf{s}(t-l)$$

DFT

Time-domain methods

$$\mathbf{y}(f,t) = \mathbf{H}(f)\ \mathbf{s}(f,t), \quad f = 1, \ldots, F$$

Solve one instantaneous ICA problem for each frequency
→ apply existing ICA techniques for instantaneous mixtures

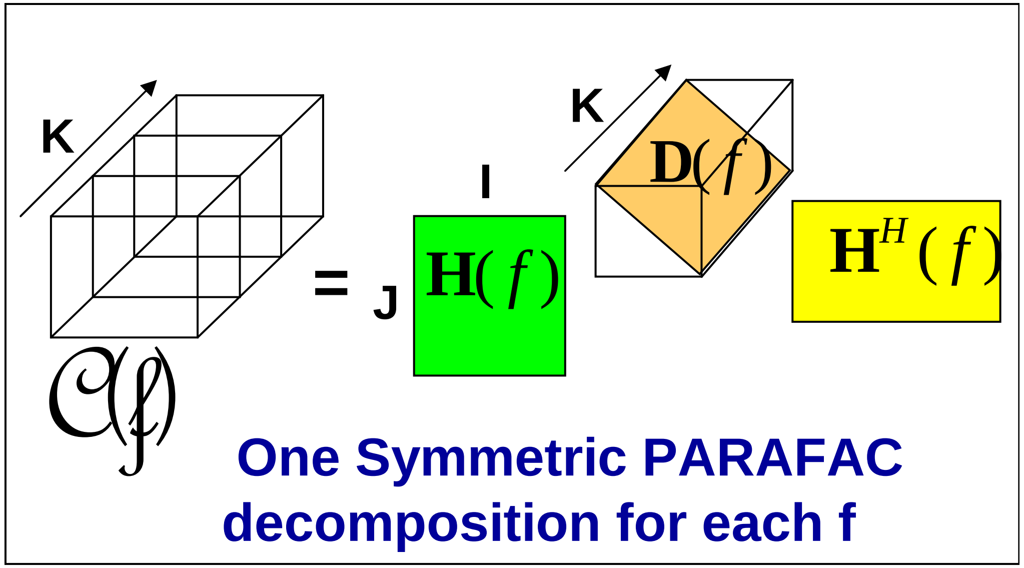
Application 5:

Blind Source Separation (convolutive mixtures)

« PARAFAC-Based Blind Separation of convolutive speech mixtures »

[Nion, Mokios, Sidiropoulos & Potamianos 2008]

$$\mathbf{y}(f,t) = \mathbf{H}(f)\,\mathbf{s}(f,t),$$
$$f = 1,\dots,F$$



One Symmetric PARAFAC decomposition for each f

Compute the F decompositions and collect **{H**(1), **H**(2), …, **H**(F)**}**

As before, works in under-determined cases

After separation stage, the job is really complete after solving:

→ arbitrary scaling and permutation of columns of **H**(f) at each frequency

→ Under-determined cases: we can not compute $\mathbf{s}(f,t) = \mathbf{H}^{\dagger}(f)\mathbf{y}(f,t)$

Application 5:

Blind Source Separation (convolutive mixtures)

« PARAFAC-Based Separation of convolutive speech mixtures »

[Nion, Mokios, Sidiropoulos & Potamianos 2008]

AUDIO DEMO: http://www.telecom.tuc.gr/~nikos/BSS_Nikos.html

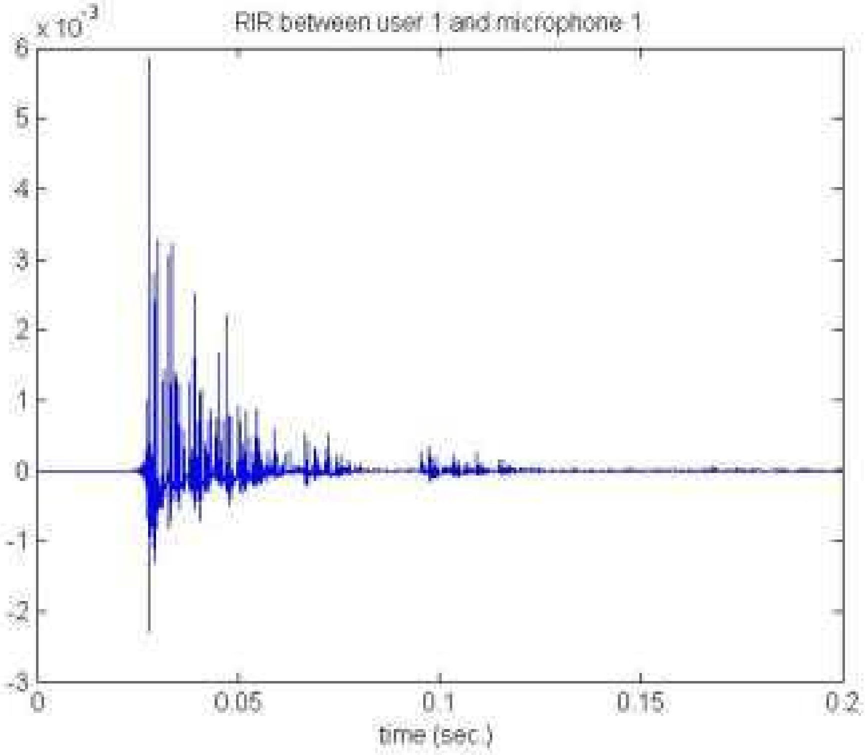
Example 1:

I=4 speech signals,

J=8 microphones

mic 1 … mic 8

$\hat{s}_1$ $\hat{s}_2$ $\hat{s}_3$ $\hat{s}_4$



Room Impulse Response (T_{60}=200 ms)

Application 5:

Blind Source Separation (convolutive mixtures)

« PARAFAC-Based Separation of convolutive speech mixtures »

[Nion, Mokios, Sidiropoulos & Potamianos 2008]

AUDIO DEMO: http://www.telecom.tuc.gr/~nikos/BSS_Nikos.html

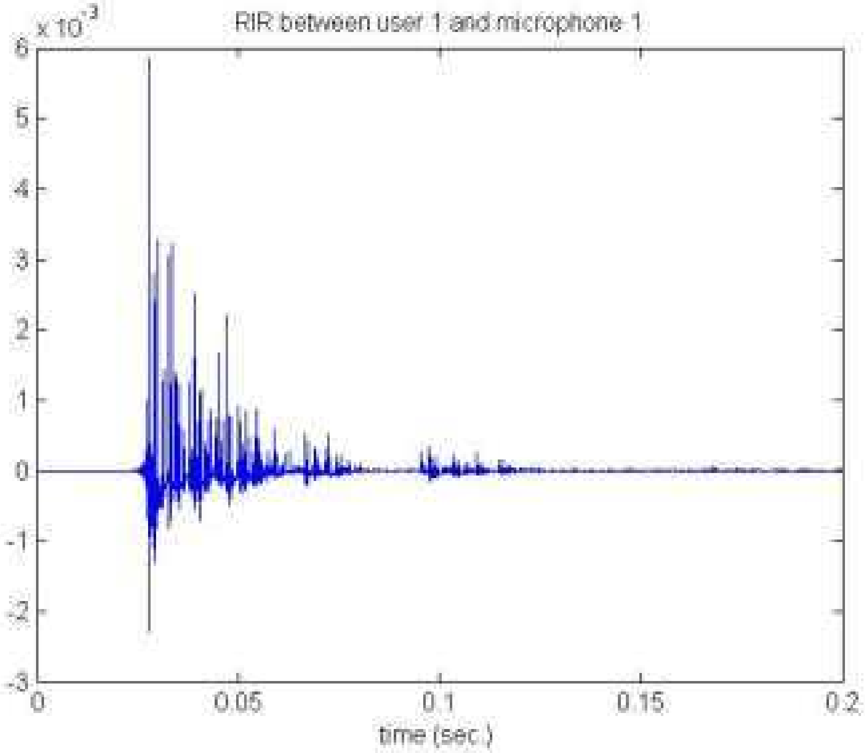
Example 2:

I=3 music signals,

J=8 microphones

mic 1 … mic 8

$\hat{s}_1$ $\hat{s}_2$ $\hat{s}_3$



Room Impulse Response (T_{60}=200 ms)

Application 6:

Target localization in MIMO radars

→MIMO radar = emerging technology.

→Principle: send orthogonal waveforms from different antennas, and capture the waveforms reflected by the targets from different receive antennas.

→ Two classes of MIMO radars: « Widely separated antennas » and « Closely spaced antennas »

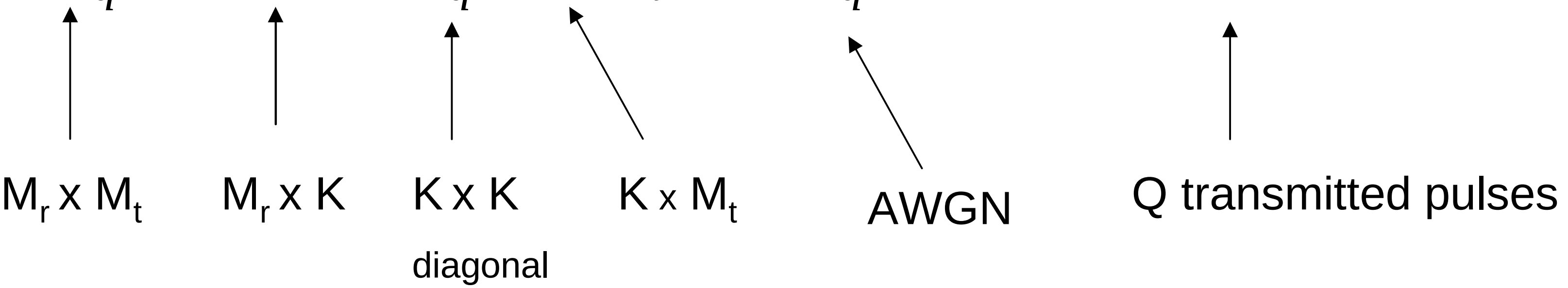
→ Exploitation of spatial diversities yields better performance (in terms of target localization, false alarm rate, …) compared to mono-antenna.

Application 6:

Target localization in MIMO radars

Data Model (after matched filtering by orthogonal transmitted pulses):

$$\mathbf{Y}_q = \mathbf{B}(\theta_r)\Sigma_q\mathbf{A}^{\mathsf{T}}(\theta_t) + \mathbf{Z}_q, \quad q = 1,\ldots,Q$$



M_r x M_t

M_r x K

K x K

diagonal

K x M_t

AWGN

Q transmitted pulses

Swerling case II target model

« Receive and Transmit steering matrices **B** and **A** are constant over the duration of Q pulses while the target reflection coefficients are varying independently from pulse to pulse».

Purpose: Localize the K targets

Application 6:

Target localization in MIMO radars

$$\mathbf{Y}_q = \mathbf{B}(\theta_r)\Sigma_q \mathbf{A}^{\mathsf{T}}(\theta_t) + \mathbf{Z}_q, \quad q = 1,\dots,Q$$

« Beamforming-based approach »: Capon estimator [Li and Stoica, 2006]

Find the (transmit,receive) angle pairs where the power $P(\theta_t,\theta_r)$ of the received signal is maximum → Compute for all possible pairs

« PARAFAC-based approach »: [Nion and Sidiropoulos, 2008]

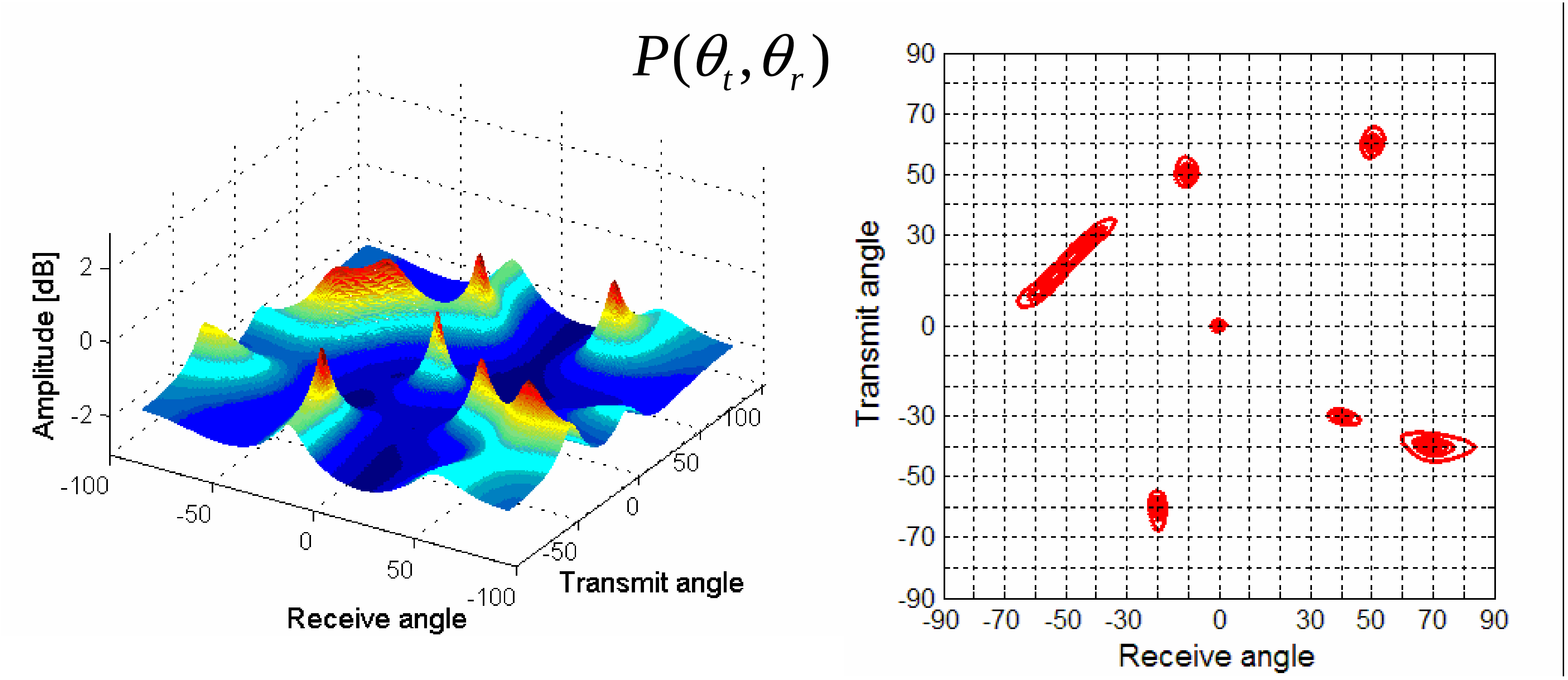
The received data model follows a deterministic PARAFAC model

→ Parametric model, find the angles from the PARAFAC decomposition

Application 6:

Target localization in MIMO radars

« Beamforming-based approach »: [Li & Stoica]



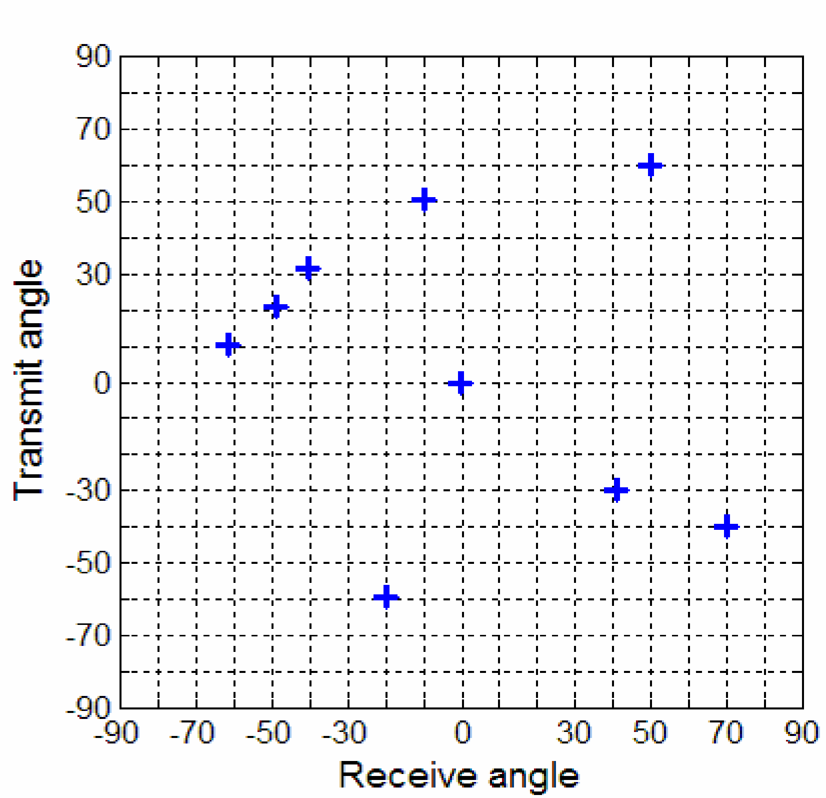
Problem: for closely spaced targets, neighboring peaks not distinguishable → detection and localization fails

Application 6:

Target localization in MIMO radars

« PARAFAC-Based Localization of multiple targets in MIMO radars»

[Nion & Sidiropoulos 2008]

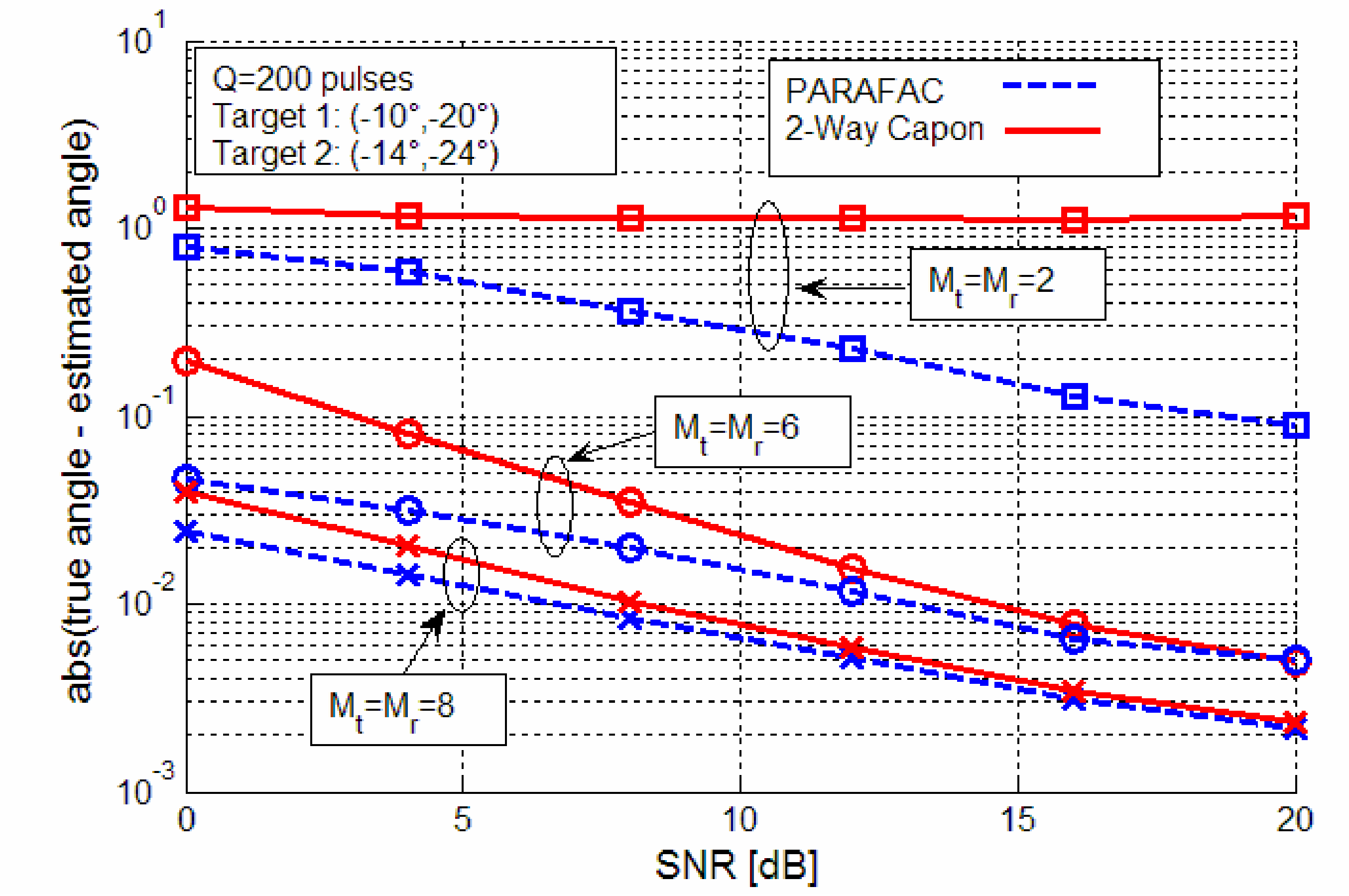


All targets are detected and localized.

Application 6:

Target localization in MIMO radars

PARAFAC vs Capon

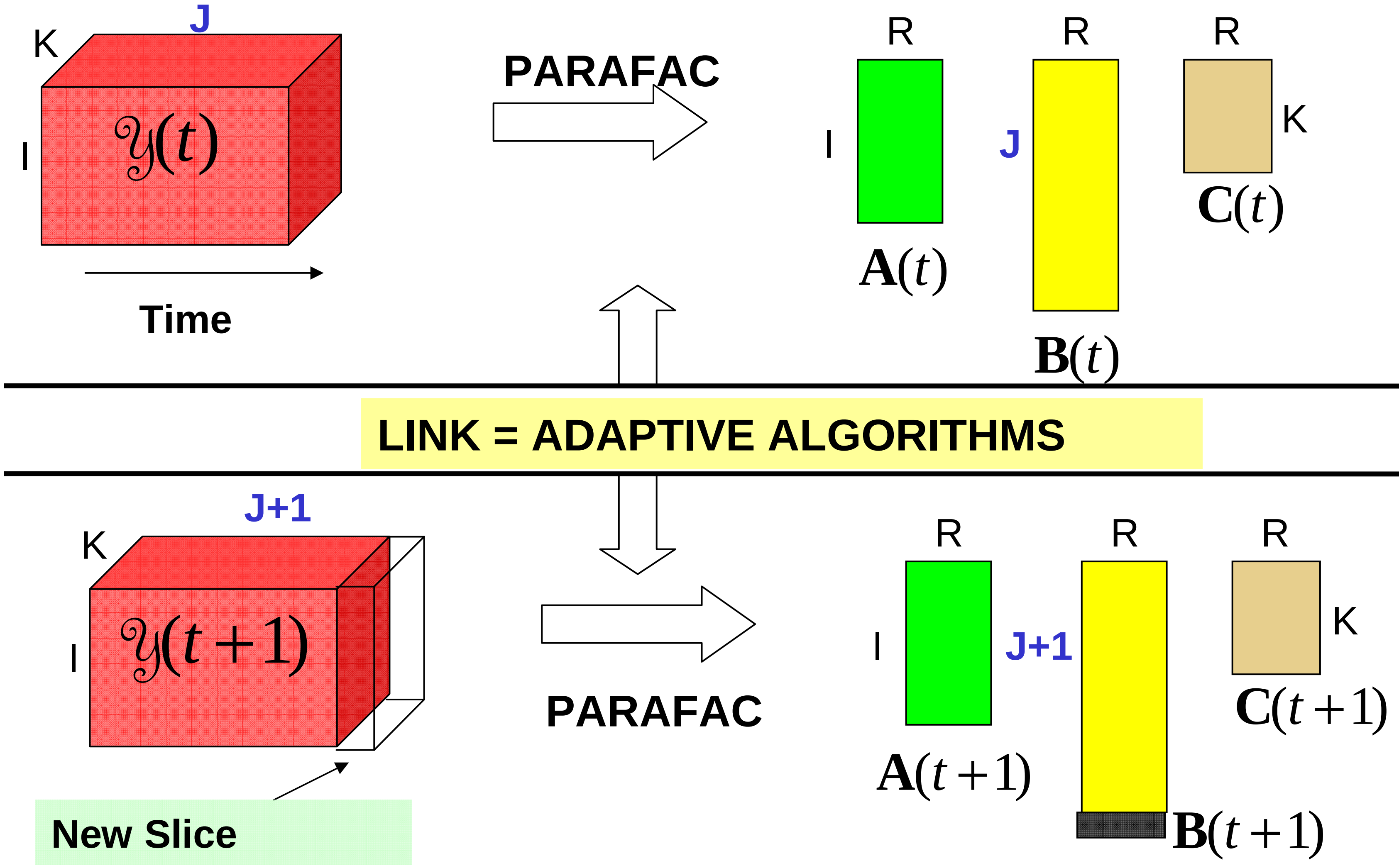


Application 7:

Tracking the PARAFAC decomposition

« Adaptive algorithms to track the PARAFAC decomposition »

[Nion & Sidiropoulos 2008]



Application 7:

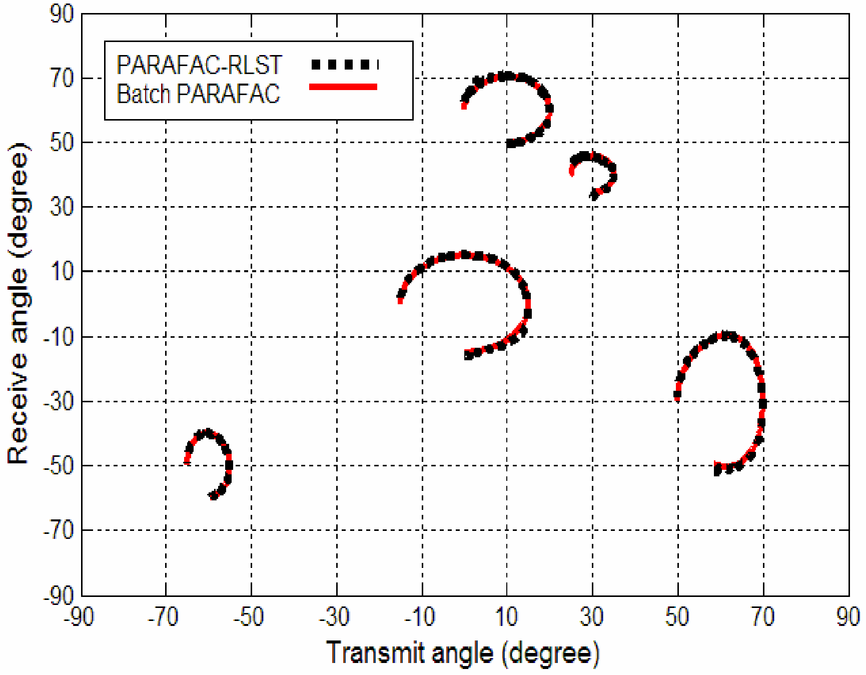
Tracking the PARAFAC decomposition

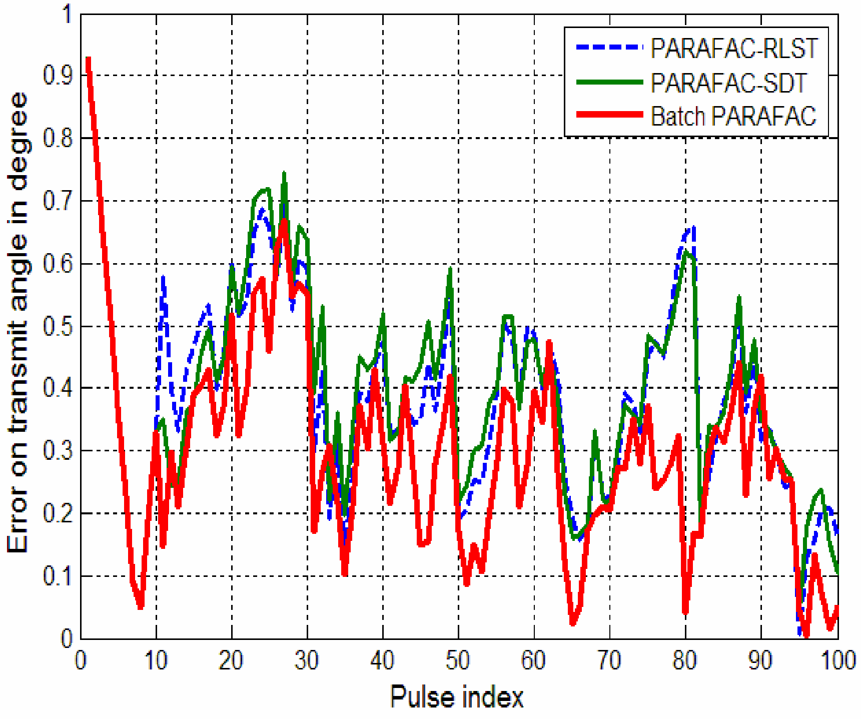
« Adaptive algorithms to track the PARAFAC decomposition »

[Nion & Sidiropoulos 2008]

Example 1: MIMO radar

5 moving targets. Estimated trajectories. Comparison between Batch PARAFAC (applied repeatedly) and PARAFAC-RLST (« Recursive Least Squares Tracking »)



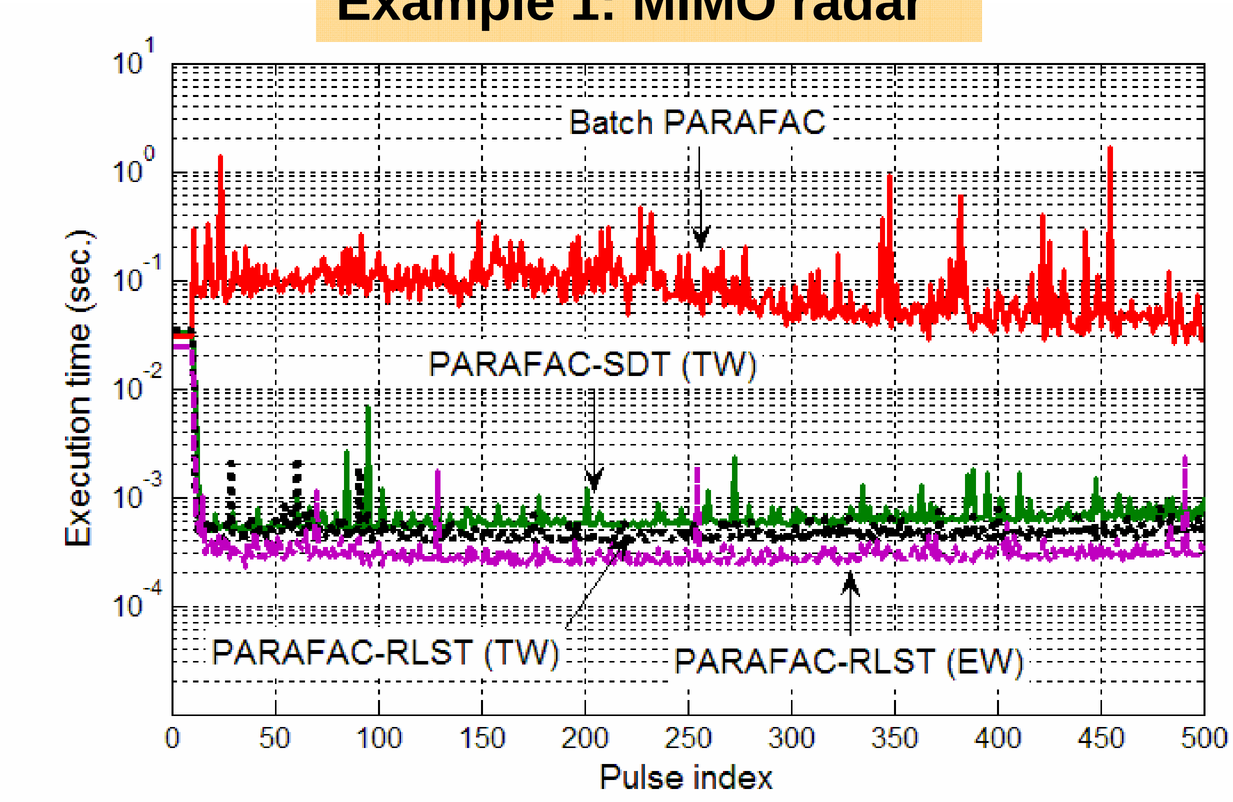


Application 7:

Tracking the PARAFAC decomposition

« Adaptive algorithms to track the PARAFAC decomposition »
[Nion & Sidiropoulos 2008]

Example 1: MIMO radar



Adaptive PARAFAC algorithms ~1000 times faster than batch ALS

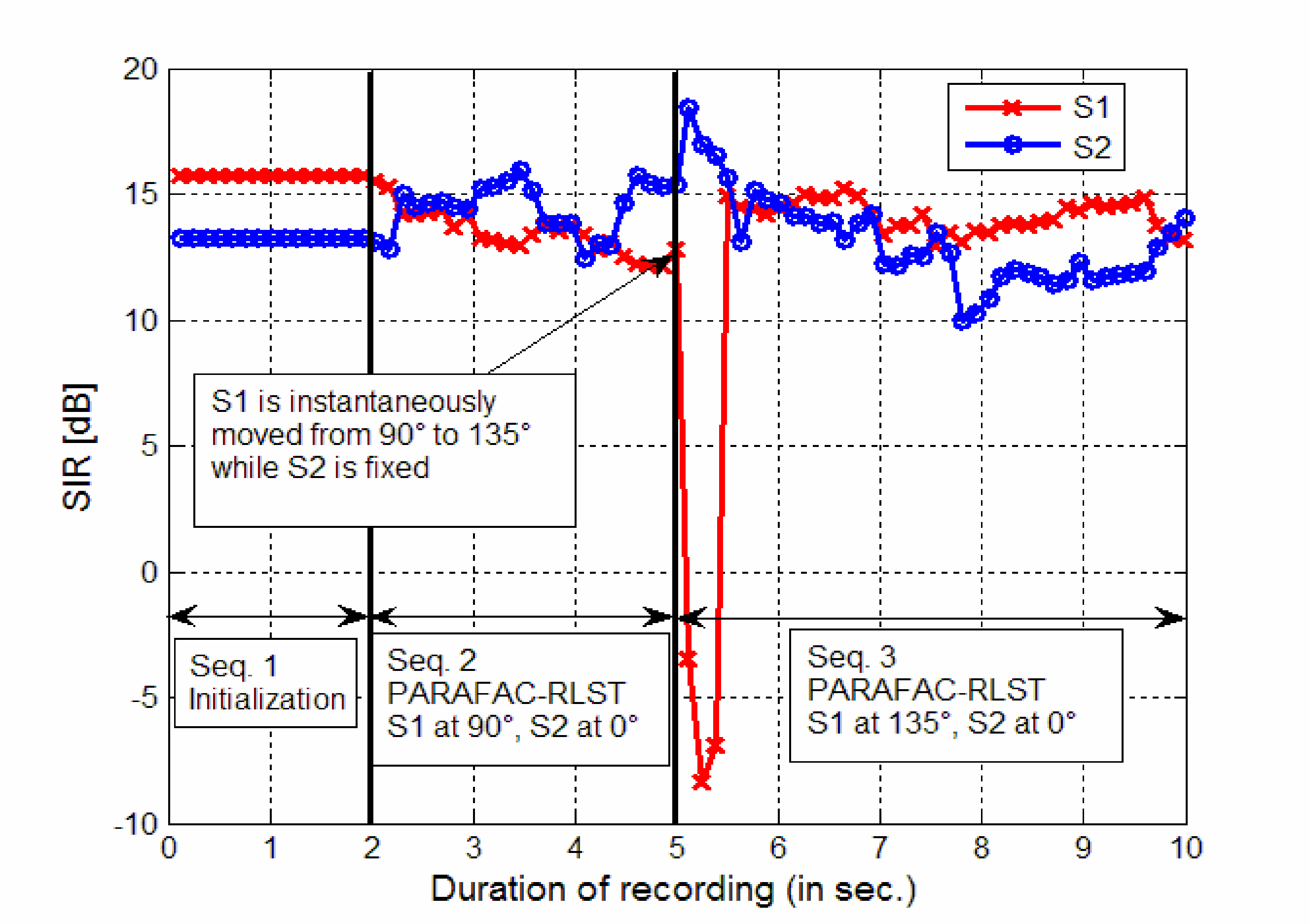
Application 7:

Tracking the PARAFAC decomposition

« Adaptive algorithms to track the PARAFAC decomposition »

[Nion & Sidiropoulos 2008]

Example 2: BSS



Conclusion

Tensor tools more powerful than matrix tools:

- More appropriate to represent and process multivariate signals (one dimension=one variable)

- Uniqueness: estimate raw data and not subspaces only

Tensor tools useful both in deterministic and statistical frameworks:

- Tensor models can represent the algebraic structure of multi-dimensional signals (e.g. CDMA signals received by multiple antennas, MIMO radars)

- Joint-Diagonalization is equivalent to symmetric PARAFAC → enjoy the benefit of PARAFAC uniqueness (even in under-determined cases) + low complexity (dimension reduction)

Many applications:

- Source separation (telecom signals, speech signals, defects analysis, …)

- Multi-Way compression and analysis (Tensor faces)

- Chemometrics

Perspectives

Towards Real-Time Tensor-Based applications:

- Adaptive PARAFAC algorithms very efficient (accurate and low complexity)

→ On chip implementation? (e.g. real-time speech separation)

- Adaptive algorithms for Block Decompositions under development

Towards New Uniqueness Bounds

- Uniqueness bounds for Block Decomposition are sufficient → find more relaxed bounds

Towards New Tensor Tools

- Develop new tensor-based (application-specific) analysis tools

Towards New Applications

- New/ Emerging applications where multi-variate data have to be represented and processed.

- Existing applications where the tensor structure was ignored until now.