The decomposition of a third-order tensor in R block-terms of rank-(L,L,1)

Model, Algorithms, Uniqueness, Estimation of R and L

Dimitri Nion & Lieven De Lathauwer

K.U. Leuven, Kortrijk campus, Belgium

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

Lieven.DeLathauwer@kuleuven-kortrijk.be

TRICAP 2009, Nurià, Spain, June 14th-19th, 2009

KATHOLIEKE UNIVERSITEIT LEUVEN

Introduction

Tensor Decompositions = Powerful **multi-linear algebra** tools that generalize matrix decompositions.

Motivation: increasing number of applications involving manipulation of multi-way data, rather than 2-way data.

Key research axes:

→ Development of new models/decompositions

→ Development of algorithms to compute decompositions

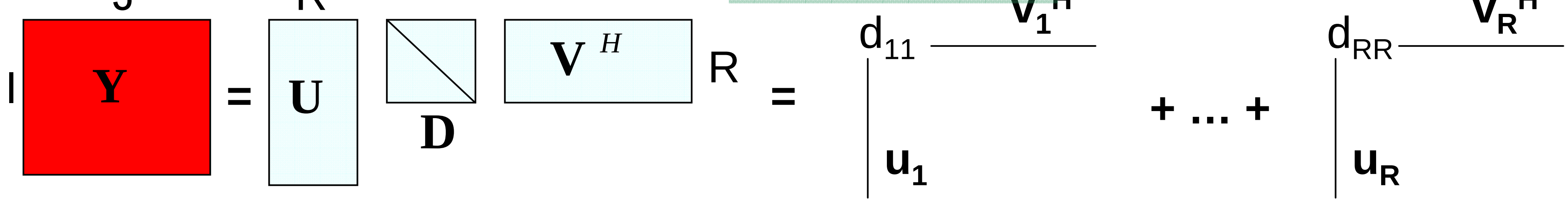
→ Uniqueness of tensor decompositions

→ Use these tools in new applications, or existing applications where the multi-way nature of data was ignored until now

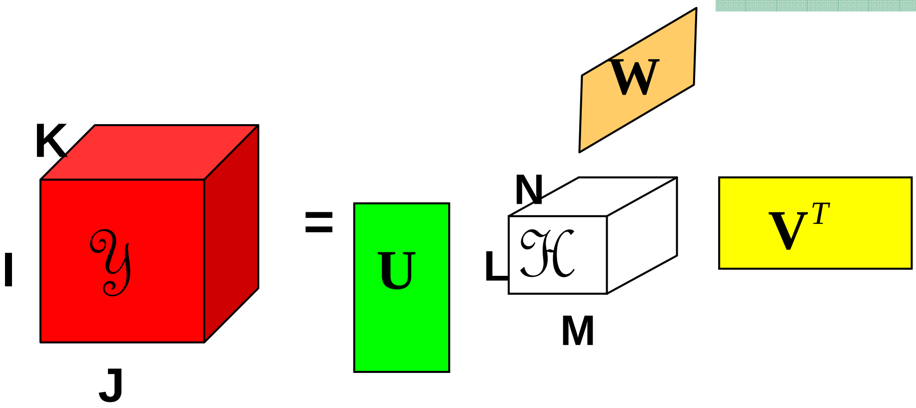
→ Tensor decompositions under constraints (e.g. imposing non-negativity or specific algebraic structures)

From matrix SVD to tensor HOSVD

Matrix SVD



Tensor HOSVD (third-order case)



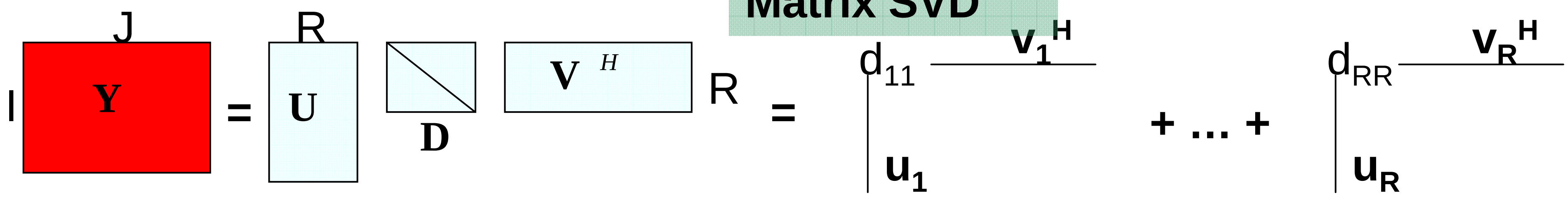
$$y_{ijk} = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} u_{il} v_{jm} w_{kn} h_{lmn}$$

$$\mathcal{Y} = \mathcal{H} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$$

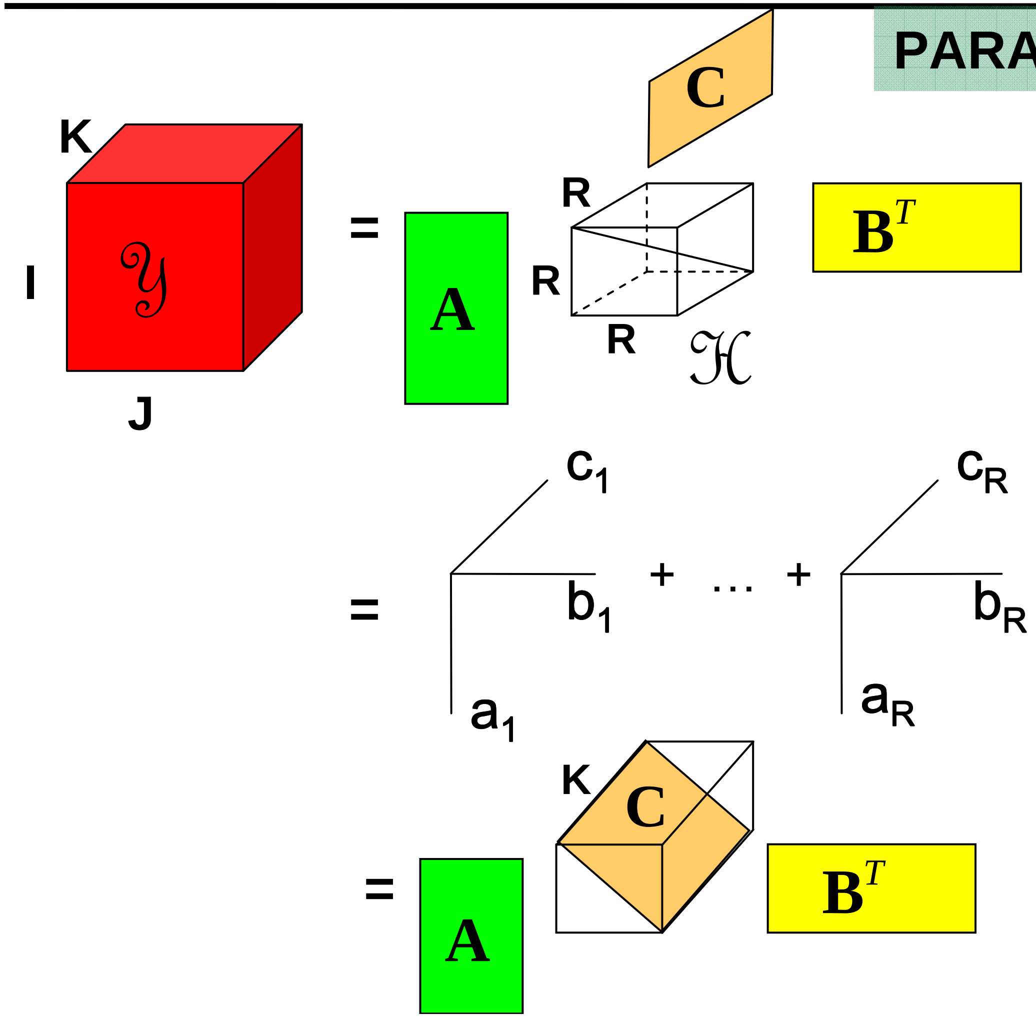
- One unitary matrix (U, V, W) per mode
- $\mathcal{H}$ is the representation of $\mathcal{Y}$ in the reduced spaces.
- We may have $L \neq M \neq N$
- $\mathcal{H}$ is **not** diagonal (difference with matrix SVD).

From matrix SVD to PARAFAC

Matrix SVD



PARAFAC decomposition



$\mathcal{H}$ is diagonal

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

Sum of R rank-1 tensors:

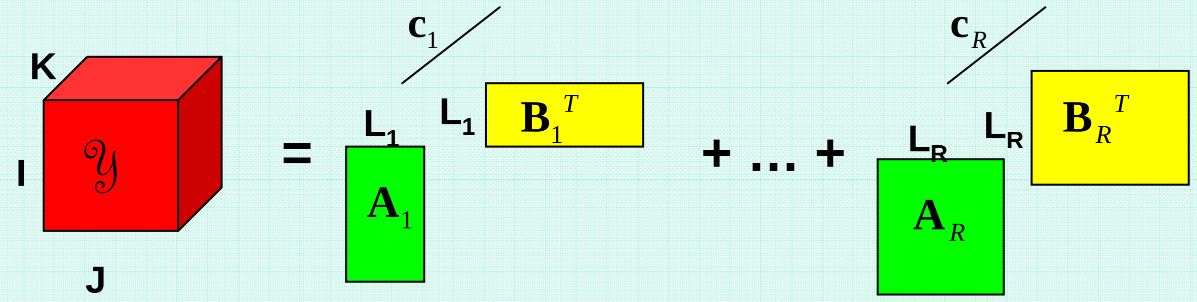
$\mathcal{Y}_1 + \ldots + \mathcal{Y}_R$

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

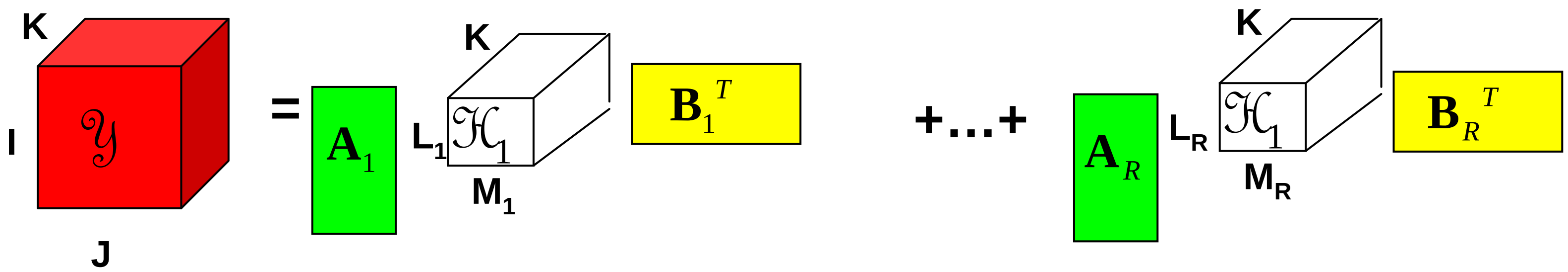
$\mathcal{Y}(:,:,k) = A \operatorname{diag}(C(k,:))\, B^T$

From PARAFAC/HOSVD to Block Components Decompositions (BCD) [De Lathauwer and Nion]

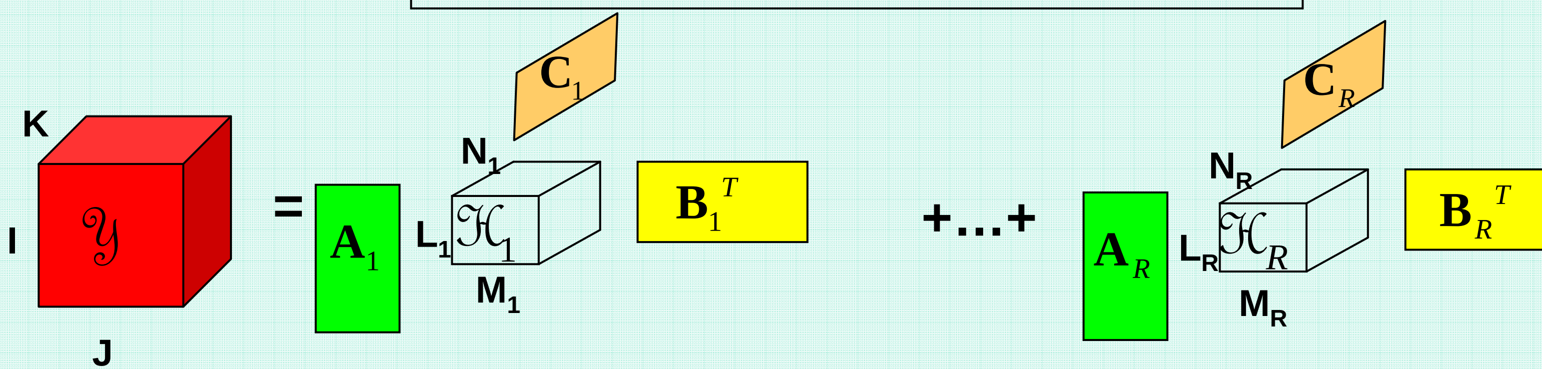
BCD in rank (L_r,L_r,1) terms



BCD in rank (L_r, M_r, .) terms

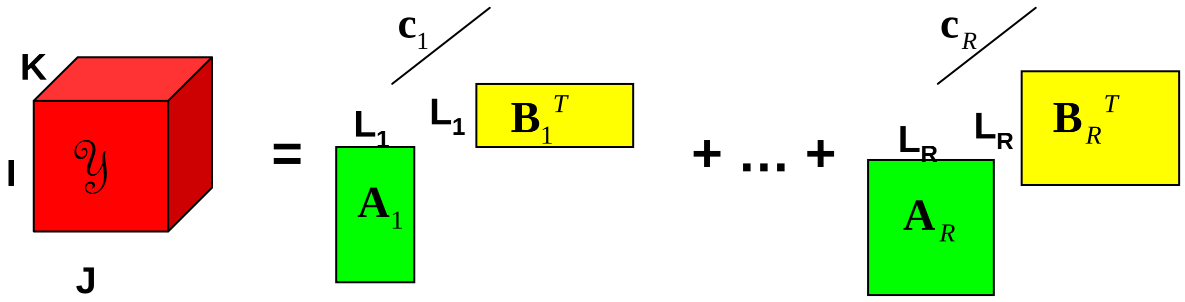


BCD in rank (L_r, M_r, N_r) terms



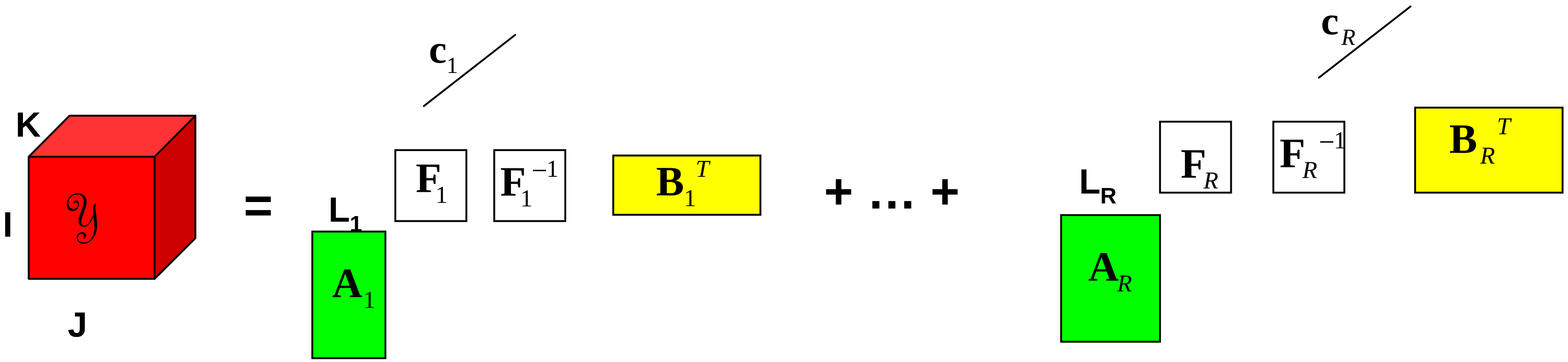
Content of this talk

BCD - (L_r,L_r,1)

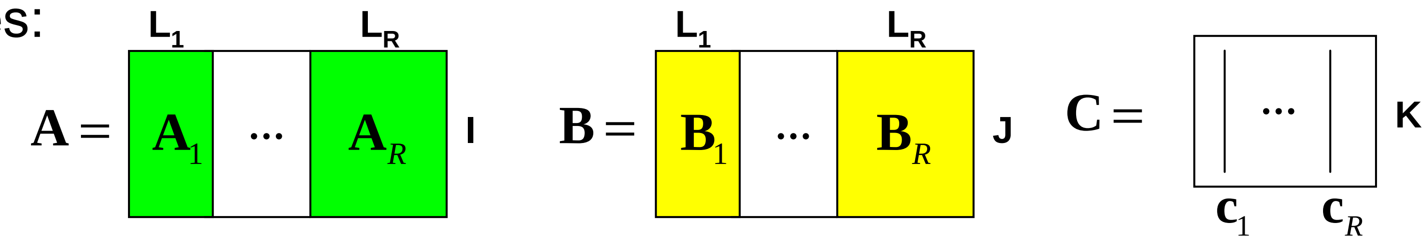


- Model ambiguities
- Algorithms
- Uniqueness
- Estimation of the parameters L_r (r = 1,…,R) and R
- An application in telecommunications

BCD - (L_r ,L_r ,1) : Model ambiguities



- Unknown matrices:



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

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

+ Each block of A and B post-multiplied by arbitrary non-singular matrix, each column of C arbitrarily scaled.

= A and B estimated up to multiplication by a **block-wise** permuted block-diagonal matrix and C by a permuted diagonal matrix.

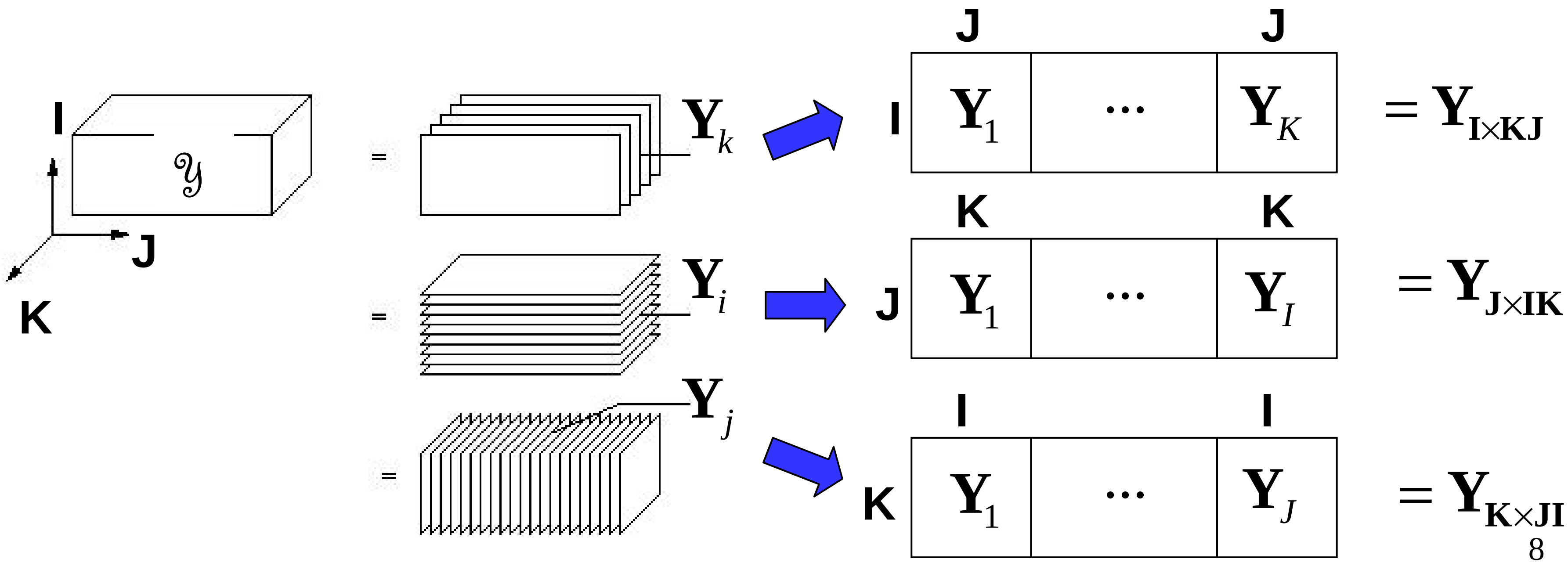
BCD - $(L_r, L_r, 1)$: Algorithms

- Usual approach: estimate **A**, **B** and **C** by minimization of

$$\Phi = \left\| \mathcal{Y} - \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^T) \circ \mathbf{c}_r \right\|_F^2 \qquad \circ = \text{outer product}$$

The model is fitted for a given choice of the parameters $\{L_r, R\}$

Exploit algebraic structure of matrix unfoldings



BCD - (L_r ,L_r ,1) : ALS Algorithm

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

$\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$ are built from 2 matrices only and have a block-wise Khatri-Rao product structure.

Initialisation : $\hat{\mathbf{A}}^{(0)}, \hat{\mathbf{B}}^{(0)}, k = 1$

$while \left|\Phi^{(k-1)} - \Phi^{(k)}\right| > \varepsilon$ (e.g. $\varepsilon = 10^{-6}$)

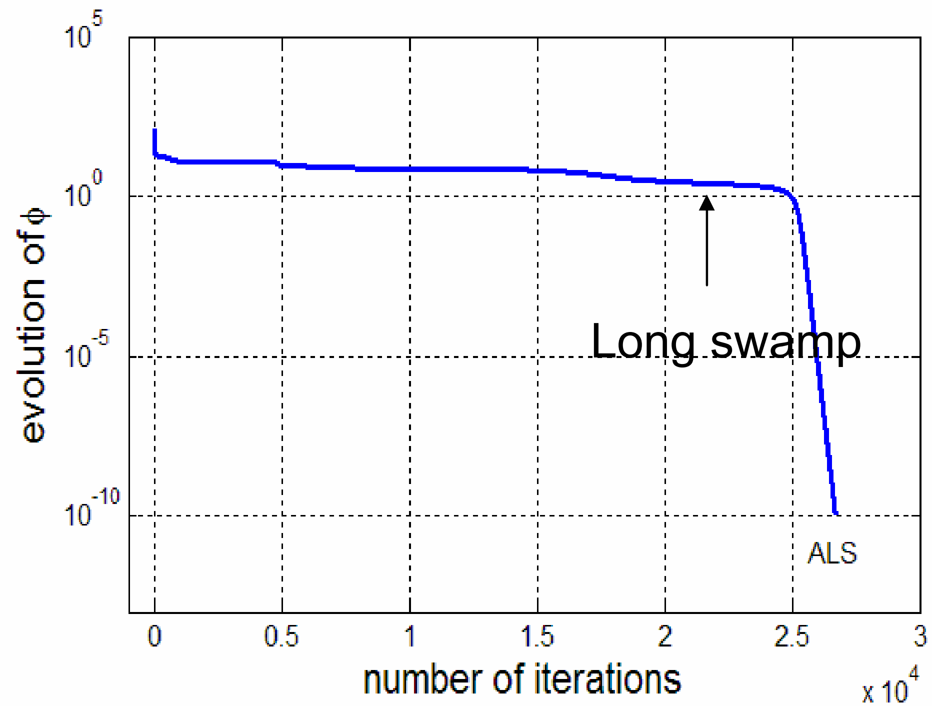
$$\hat{\mathbf{C}}^{(k)} = \mathbf{Y}_{\mathbf{K}\times\mathbf{JI}} \cdot \left[\mathbf{Z}_1(\hat{\mathbf{B}}^{(k-1)}, \hat{\mathbf{A}}^{(k-1)})\right]^{\dagger} \quad (1)$$

$$\hat{\mathbf{B}}^{(k)} = \mathbf{Y}_{\mathbf{J}\times\mathbf{IK}} \cdot \left[\mathbf{Z}_2(\hat{\mathbf{A}}^{(k-1)}, \hat{\mathbf{C}}^{(k)})\right]^{\dagger} \quad (2)$$

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

$k \leftarrow k + 1$

ALS algorithm: problem of swamps



← 27000 iterations !

Observation:

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

<u>Long Swamps typically occur when:</u>

- **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 ALS

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

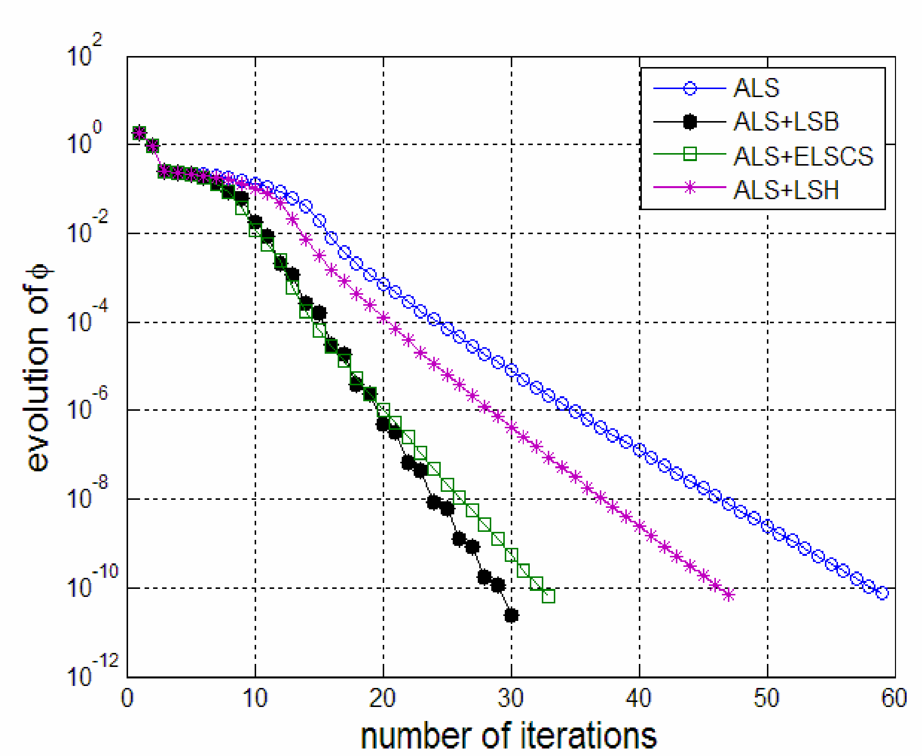
Alternate update of m and θ :

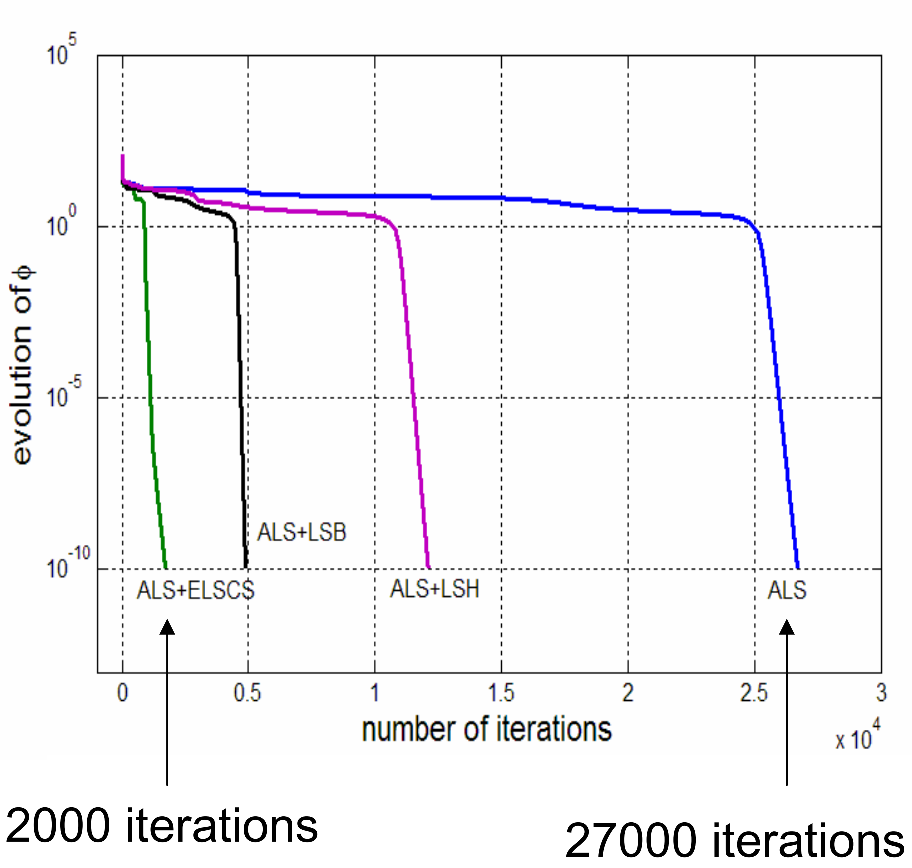
Update m : for θ fixed, $\dfrac{\partial \Phi(m,\theta)}{\partial m} =$ 5th order polynomial in m

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

Improvement 1 of ALS: Line Search

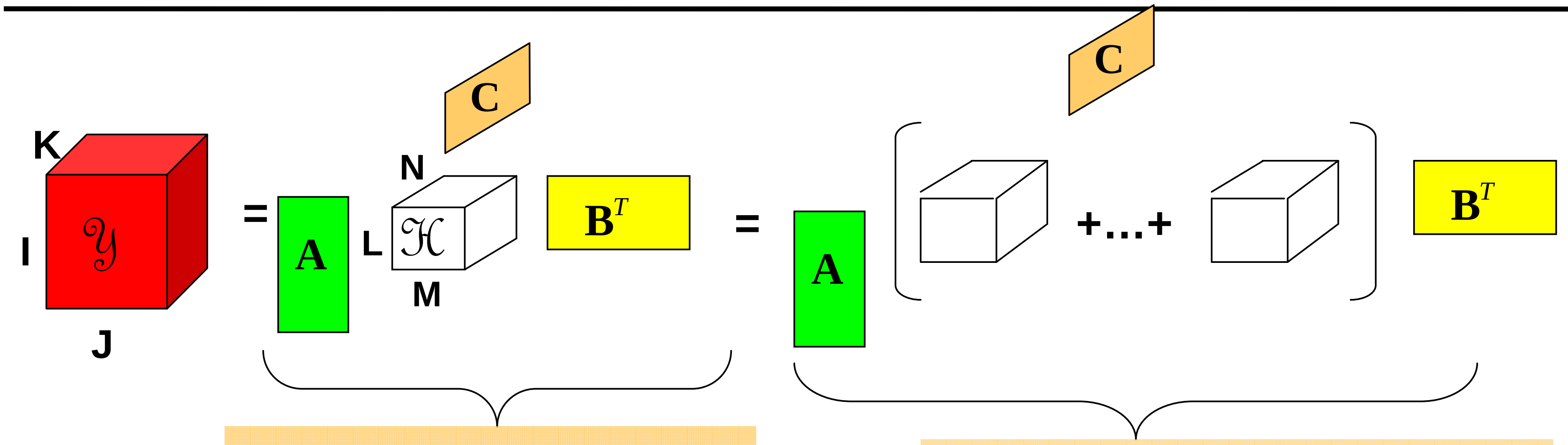
«easy» problem	«difficult» problem





➤ ELS → Large reduction of the number of iterations at a very low additional complexity w.r.t. standard ALS

Improvement 2 of ALS: Dimensionality reduction



STEP 1:

HOSVD of $\mathcal{Y}$

STEP 2:

BCD of the small core tensor $\mathcal{H}$ (compressed space)

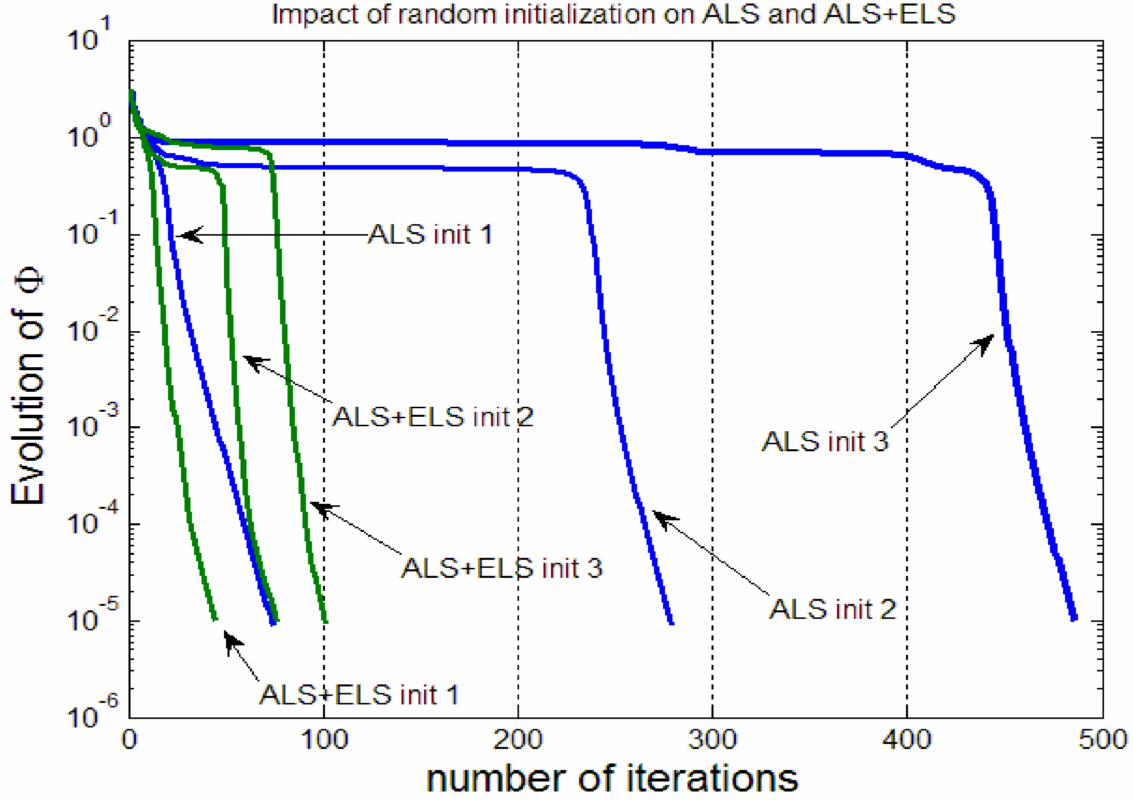
STEP 3:

Come back to original space

+ a few refinement iterations in 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 ?

YES For the BCD-(L,L,1), if A and B are full column rank (so I and J have to be long enough), there is an easy way to find a good intialization, in same spirit as Direct Trilinear Decomposition (DTLD) used to initialize PARAFAC (not detailed in this talk)**.**

Other algorithms

Existing algorithms for PARAFAC can be adapted to Block-Component-Decompositions. Examples:

- Levenberg-Marquardt algorithm (Gauss-Newton type method),
- Simultaneous Diagonalization (SD) algorithms → **let's say a few words on this technique.**

SD for PARAFAC (De Lathauwer, 2006)

- Initial condition to reformulate PARAFAC in terms of SD: $\min(IJ, K) \geq R$
- PARAFAC decomposition can be computed by solving a SD problem:

$$\mathbf{M}_n = \mathbf{W}\mathbf{D}_n\mathbf{W}^T, \ \text{n=1,...,R}, \ \mathbf{D}_n \ \text{is R} \times \text{R diagonal}$$

- Advantage: Low complexity (only R matrices of size RxR to diagonalize + direct use of existing fast algorithms designed for SD)
- SD reformulation yields a uniqueness bound generically more relaxed than Kruskal bound

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

BCD - (L ,L ,1) : computation via Simultaneous Diag.

(Nion & De Lathauwer, 2007)

- Results established for BCD-(L,L,1), i.e., same L for the R terms
- Initial condition to reformulate BCD-(L,L,1) in terms of SD: $\min(IJ, K) \geq R$
- Then the decomposition can be computed by solving a SD problem:

$$\mathbf{M}_n = \mathbf{W}\mathbf{D}_n\mathbf{W}^T, \ \text{n=1,...,R}, \ \mathbf{D}_n \ \text{is } \mathrm{R} \times \mathrm{R} \ \text{diagonal}$$

- Advantage: Low complexity (only R matrices of size RxR to diagonalize + direct use of existing fast algorithms designed for SD)
- SD reformulation yields a new, more relaxed uniqueness bound (next slide)

BCD - (L ,L ,1) : Uniqueness

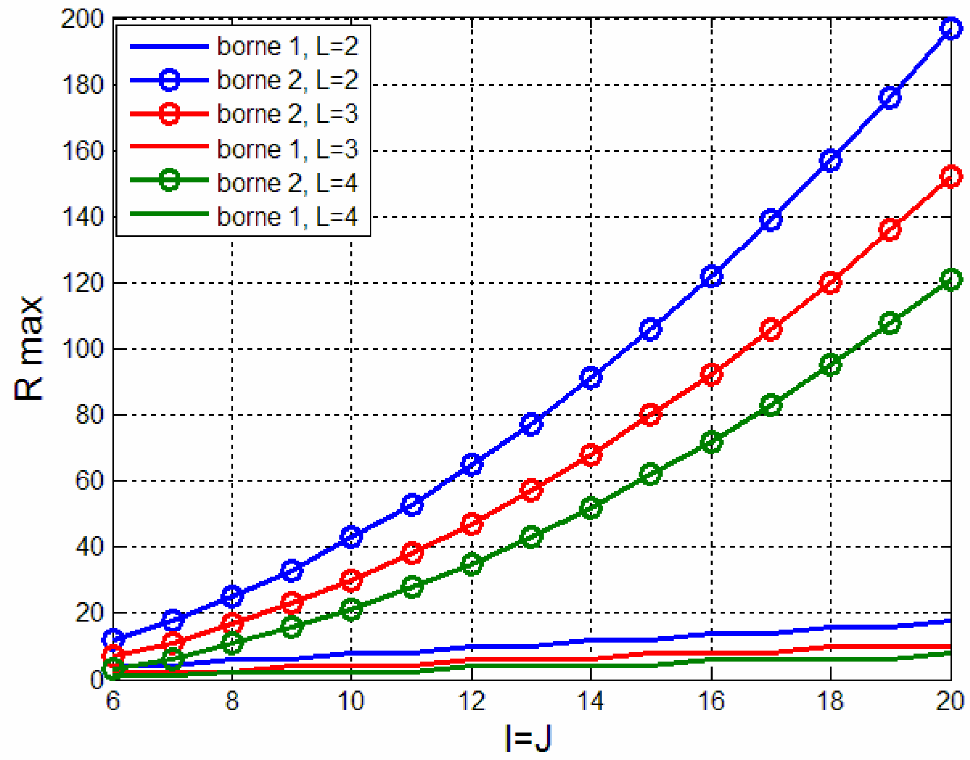
(Nion & De Lathauwer, 2007)

Sufficient bound 1
[De Lathauwer 2006]

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

Sufficient bound 2
[Nion & De Lathauwer, 2007] :

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



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

New Bound much more relaxed

Concluding remarks on algorithms

→ Standard ALS sometimes slow (swamps)

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

→ Levenberg-Marquardt → convergence very fast, less sensitive to ill-conditioned data, but higher complexity and memory (dimensions of Jacobian matrix=IJK)

→ Simultaneous diagonalization: a very attractive algorithm (low complexity and good accuracy).

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

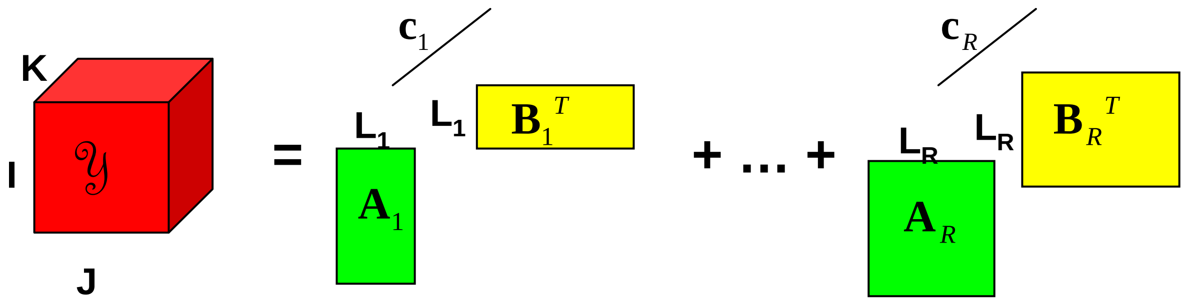
- Dimensionality reduction pre-processing step (e.g. via Tucker/HOSVD)
- Find a good initialization if possible.

→ 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, … (e.g. in Telecoms Applications)
- non-negativity constraints (e.g. Chemometrics applications)

BCD - (L_r ,L_r ,1) : estimation of R and L_r

Problem: Given a tensor $\mathcal{Y}$, how to estimate the number of terms R and the rank L_r of the matrices $\mathbf{A}_r$ and $\mathbf{B}_r$ that yield a reasonable (L_r, L_r, 1) model?



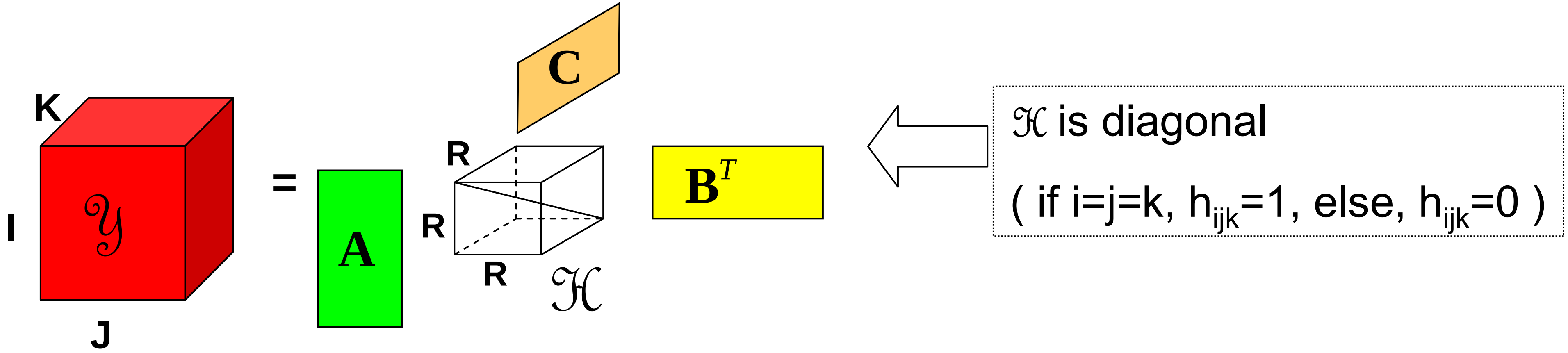
- **Criterion 1:** Simple approach: examinate singular values of matrix unfoldings.

  - **Y** (JIxK) generically rank R if $\min(JI, K) \geq R$
  - **Y** (IKxJ) generically rank $N = \sum_{r=1}^{R} L_r$ if $\min(IK, J) \geq N$
  - **Y** (KJxI) generically rank N if $\min(KJ, I) \geq N$

- If noise level not too high and if conditions on dimensions satisfied, the number of significant singular values yields an estimate for R and/or N.

CORCONDIA (Core Consistency Diagnostic)

Core idea: PARAFAC can be seen as a particular case of Tucker model, where the core tensor is diagonal.

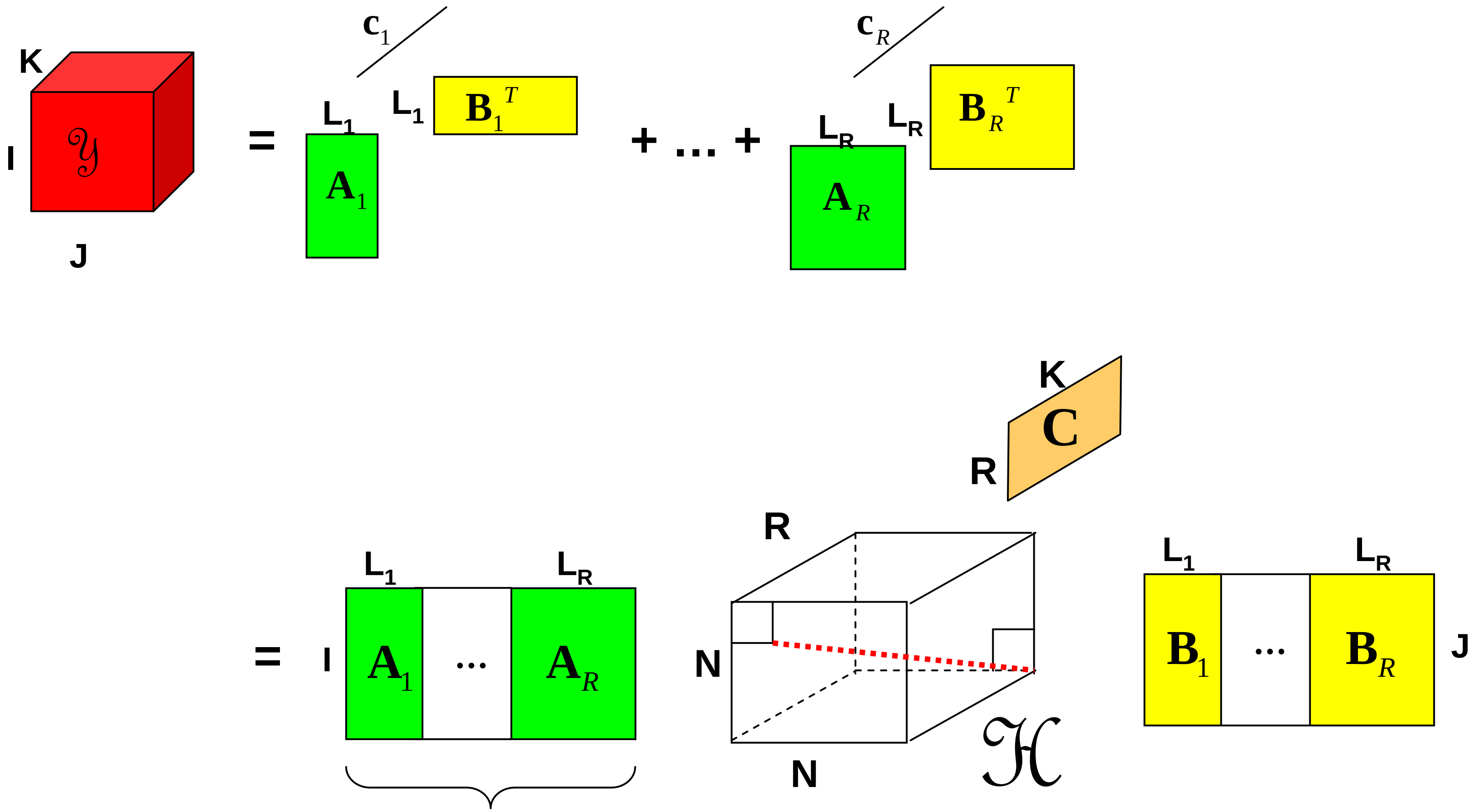


Method [Bro et al.]

- Choose a set of plausible values for R.
- For a given test (i.e., for a given R), fit a PARAFAC model and compute the Least Squares estimate of the core tensor $\mathcal{H}$,
- and measure the diagonality of the core tensor: $C = 100\,(1 - \frac{\|\mathcal{H} - \hat{\mathcal{H}}\|_F^2}{R})$
- Examinate the core consistency measurements to select R

Block-(L_r ,L_r ,1) CORCONDIA

Core idea: BCD-(L_r ,L_r , 1) can be seen as a particular case of Tucker model, where the core tensor is « block-diagonal ».



$$N = \sum_{r=1}^{R} L_r$$

Block-(L_r ,L_r ,1) CORCONDIA

Criterion 2: So we can proceed in a way similar to CORCONDIA for PARAFAC

- Choose a set of plausible values for R and L_r , r=1,…,R.

- For a given test (i.e., for given R and L_r 's), fit a BCD-(L_r ,L_r ,1) model and compute the Least Squares estimate of the core tensor $\mathcal{H}$,

- and measure the block - diagonality of the core tensor:

$$C_{COR} = 100 \; (1 - \frac{\left\| \mathcal{H} - \hat{\mathcal{H}} \right\|_F^2}{RL})$$

- Examinate the multiple core consistency measurements to select the most plausible parameters

---

Criterion 3: Similarly to PARAFAC, better to couple Block-CORCONDIA to other criteria, e.g., examination of the relative Fit to the (L_r , L_r, 1) model:

$$C_{Fit} = 100 \; (1 - \frac{\left\| \mathcal{Y} - \hat{\mathcal{Y}} \right\|_F^2}{\left\| \mathcal{Y} \right\|_F^2})$$

Block-(L_r ,L_r ,1) CORCONDIA

- **Example 1:** I=12, J=12, K=50, L=2, R=3 ($L=L_1=L_2=L_3$)

Complex data (random), and SNR=10 dB

Test: R_{try} = {1,2,3,4,5,6} and L_{try}={1,2,3,4}

Note: For each (R,L) pair, the decomposition is computed via ALS+ELS algorithm and 5 different starting points.

C_{Fit}= (rows: R_{try} ↓, columns: L_{try} →)

22.3	36.6	38.1	39.3
38.6	66.6	67.8	69.1
56.3	**91.2**	91.3	91.4
71.4	91.5	91.7	91.8
84.1	91.7	91.9	92.1
91.5	92.0	92.3	92.4

C_{COR}= (rows: R_{try} ↓, columns: L_{try} →)

100	100	100	100
99	99.8	< 0	< 0
98.9	**99.4**	< 0	< 0
84.8	30.9	< 0	< 0
< 0	< 0	< 0	< 0
< 0	< 0	< 0	< 0

→ **L=2 and R=3 corresponds to the intersection of the acceptable values of Fit and the ones for Core Consistency.**

Block-(L_r ,L_r ,1) CORCONDIA

- **Example 2:** I=12, J=12, K=50, L=3, R=3 ($L=L_1=L_2=L_3$)

Complex data (random), and SNR=10 dB

Test: R_{try} = {1,2,3,4,5,6} and L_{try}={1,2,3,4,5}

C_{Fit}= (rows: R_{try} ↓, columns: L_{try} →)

20.3	32.8	38.1	40.4	41.6
37.8	60.8	68.4	69.8	70.4
54.2	**81.3**	**91.4**	91.4	91.5
68.7	88.1	91.7	91.8	91.9
78.1	91.4	91.9	91.1	92.2
82.8	91.9	92.3	92.5	92.6

C_{COR}= (rows: R_{try} ↓, columns: L_{try} →)

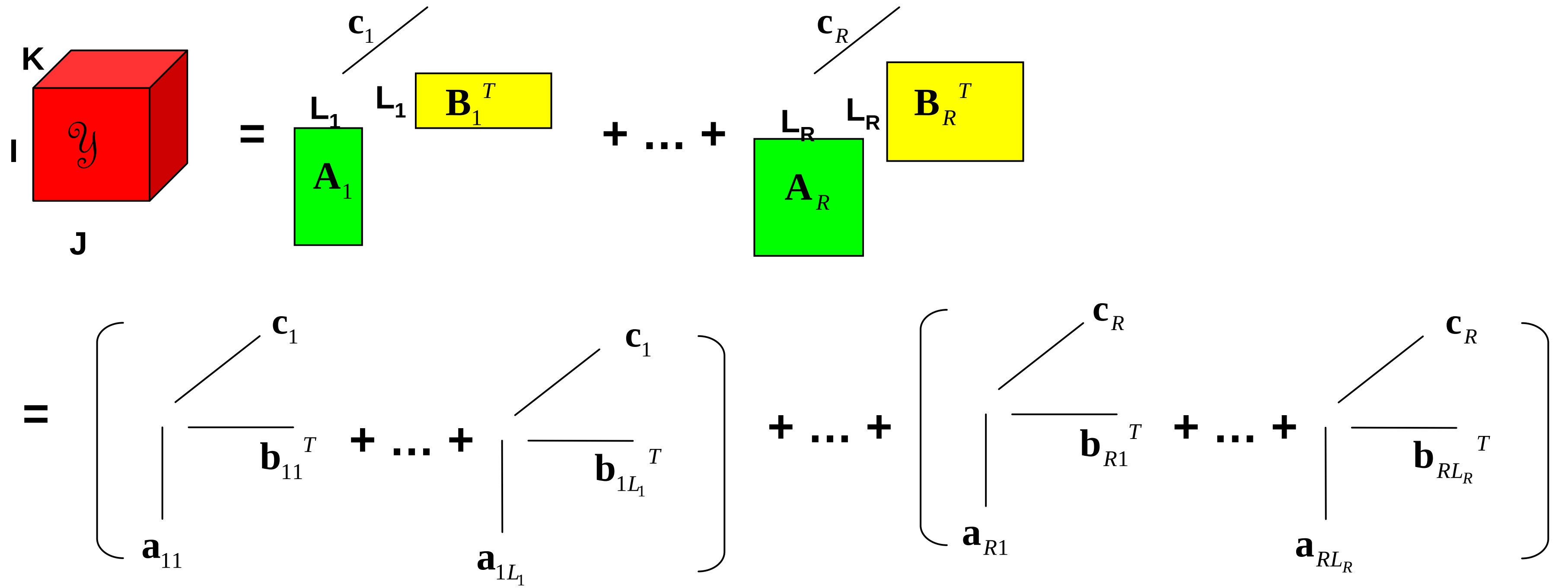
100	100	100	100	100
95.2	96.1	55.1	< 0	< 0
94.1	**64.2**	**59.9**	< 0	< 0
60.3	< 0	< 0	< 0	< 0
< 0	< 0	< 0	< 0	< 0
< 0	< 0	< 0	< 0	< 0

→(R,L)=(3,2) and (R,L)=(3,3) could be chosen.

→ Find with other criteria to help in the final decision

Block-(L_r ,L_r ,1) CORCONDIA

- **Criterion 4:** use the BCD-(L,L,1) structure



- Can be seen as PARALIND (Parallel profiles with Linear Dependencies) [Bro, Harshman, Sidiropoulos]

- Repetition of the vectors $\mathbf{c}_r$ in each term.

- **Idea:** fit a rank-N PARAFAC model (N is the number of rank-1 terms) and compute correlation of estimated **c** vectors

Block-(L_r ,L_r ,1) CORCONDIA

- From example 2, ambiguous choice: (R,L)=(3,2) or (R,L)=(3,3) ?
- Fit a rank-6 and a rank-9 PARAFAC model and check if the pairing of the estimated **c** vectors clearly appears

1	0.15	0.99	0.09	0.14	0.86
0.15	**1**	0.15	0.39	0.95	0.41
0.99	0.15	1	0.10	0.13	0.86
0.09	0.39	0.10	**1**	0.24	0.12
0.13	**0.95**	0.13	0.24	1	0.45
0.86	0.41	0.86	**0.12**	0.45	1

Clustering in R=3 groups of 2 vectors « not good »

1	0.17	0.17	0.18	0.11	0.09	0.11	0.99	0.99
0.17	**1**	0.99	0.99	0.10	0.12	0.10	0.17	0.18
0.17	**0.99**	1	0.99	0.10	0.11	0.10	0.17	0.18
0.18	**0.99**	0.99	1	0.13	0.14	0.13	0.18	0.19
0.11	0.10	0.10	0.13	1	**0.99**	0.99	0.12	0.13
0.09	0.12	0.11	0.14	0.99	**1**	0.99	0.10	0.11
0.11	0.10	0.10	0.13	0.99	**0.99**	1	0.12	0.13
0.99	0.17	0.17	0.18	0.12	0.10	0.12	1	0.99
0.99	0.18	0.18	0.19	0.13	0.11	0.13	0.99	1

Clustering in R=3 groups of 3 vectors « good »

An application of the BCD-(L_r ,L_r ,1):

Blind Source Separation in telecommunications

CDMA (« Code Division Multiple Access ») signals

→ Used in 3rd generation wireless standard (UMTS)

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

User 1 wants to transmit $\mathbf{s}_1$=[1 -1 -1].

→ CDMA code allocated to user 1: $\mathbf{c}_1$=[1 -1 1 -1].

→ User 1 transmits [+ $\mathbf{c}_1$ - $\mathbf{c}_1$ - $\mathbf{c}_1$]

→ User 2 transmits his symbols spread by his own CDMA code $\mathbf{c}_2$, orthogonal to $\mathbf{c}_1$, etc

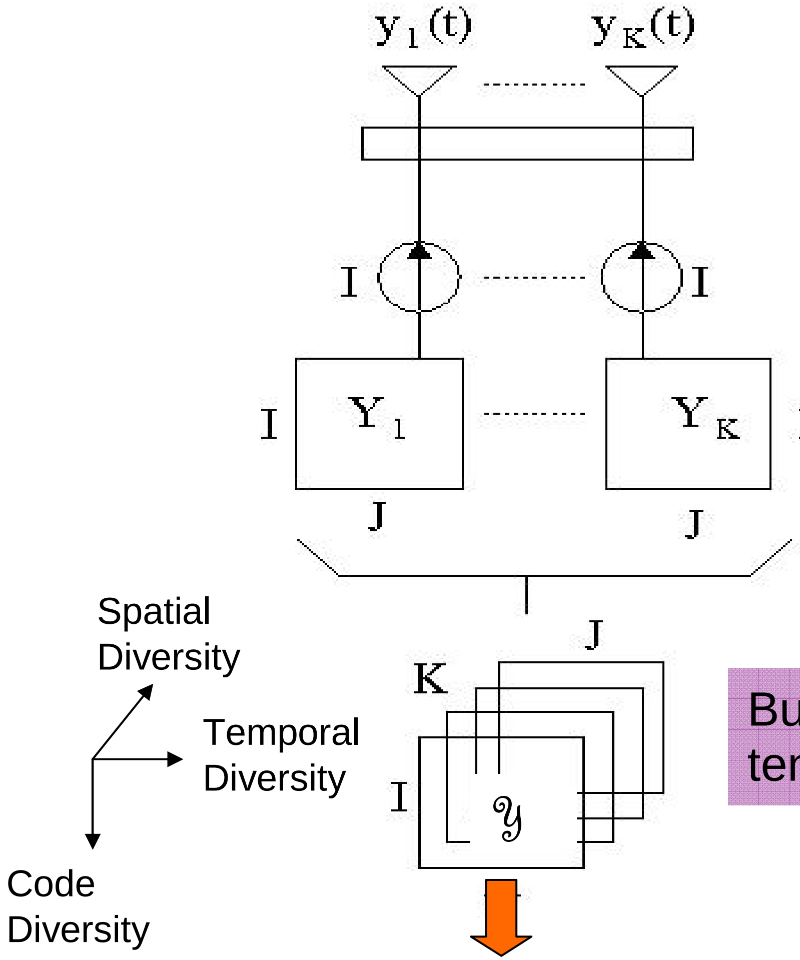
Signals received by an antenna array.

Signal received by each antenna = mixture of signals transmitted by users, affected by wireless channel effects.

Purpose: Separate these signals, from exploitation of the received signals only.

An application of the BCD-(L_r ,L_r ,1):

Blind Source Separation in telecommunications



K receive antennas

Chip rate sampling (I times faster than symbol rate)

Observation during J symbol periods

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

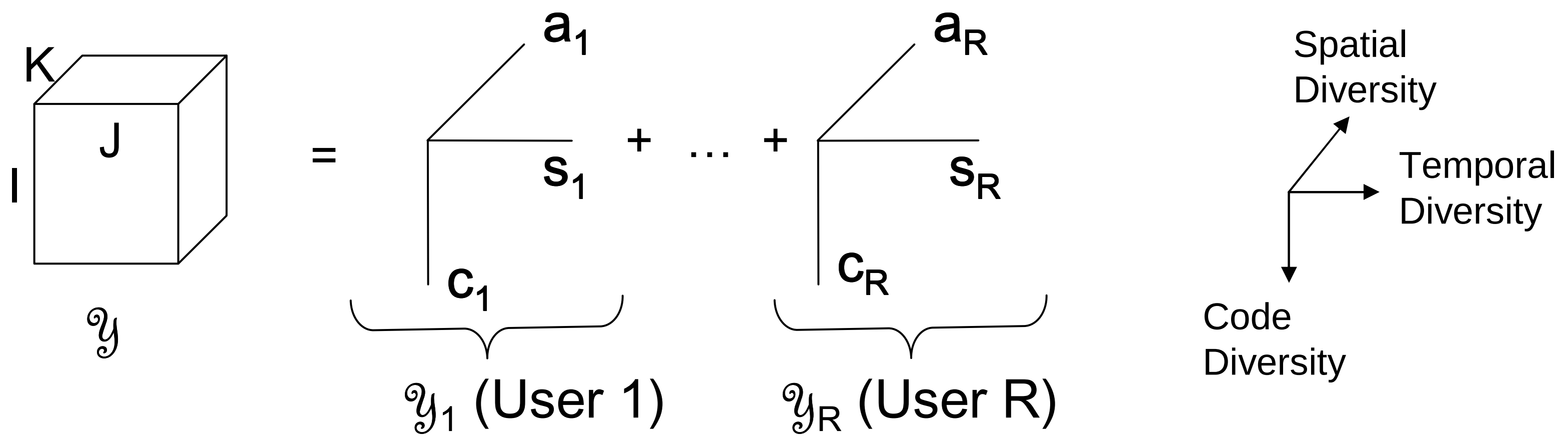
Decompose $\mathcal{Y}$ to blindly estimate the transmitted symbols.
Which decomposition to use? → the one that best reflects the algebraic structure of the data

An application of the BCD-(L_r ,L_r ,1):

Blind Source Separation in telecommunications

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

Use PARAFAC [Sidiropoulos et al.]



I = length of the CDMA codes

J = number of symbols

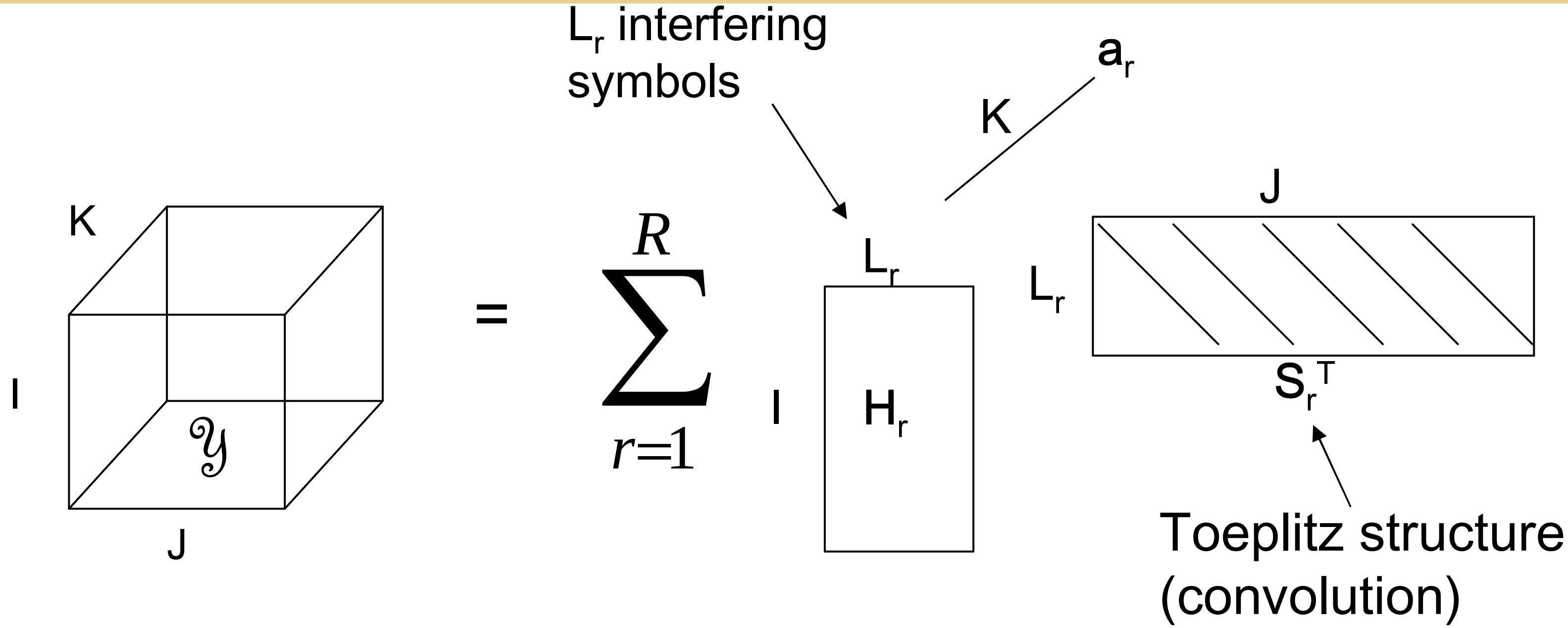
K = number of antennas at the receiver

« 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.

An application of the BCD-(L_r ,L_r ,1):

Blind Source Separation in telecommunications

Case 2: Multi-path propagation with inter-symbol-interference but far-field reflections only. Use PARALIND [Sidiropoulos & Dimic] or BCD-(L,L,1) [De Lathauwer & de Baynast]



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

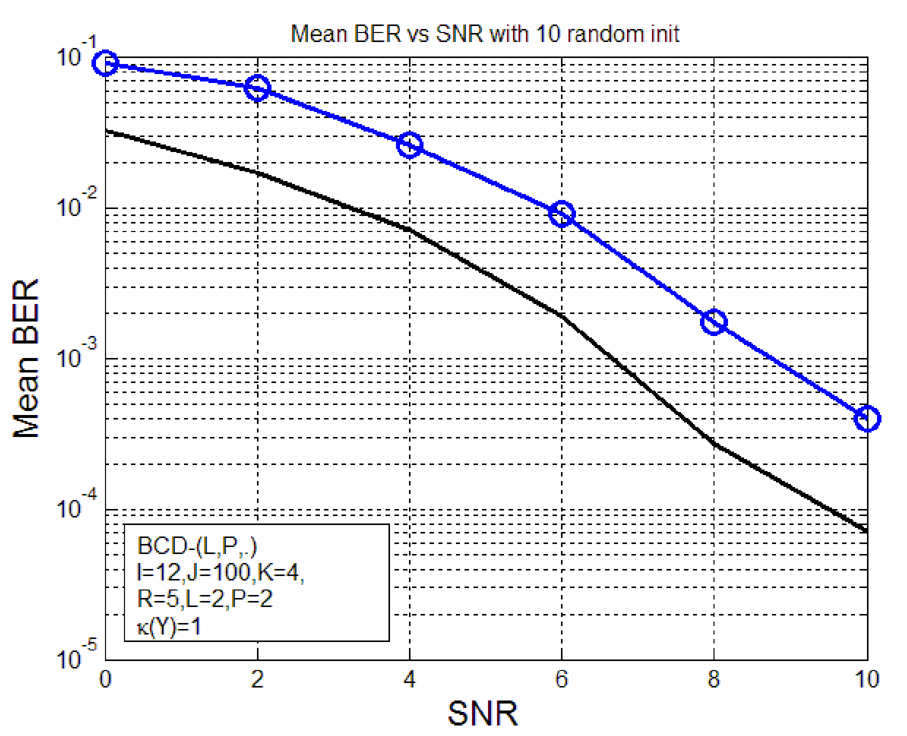
S_r → Symbol matrix, holds the J symbols of interest for user r

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

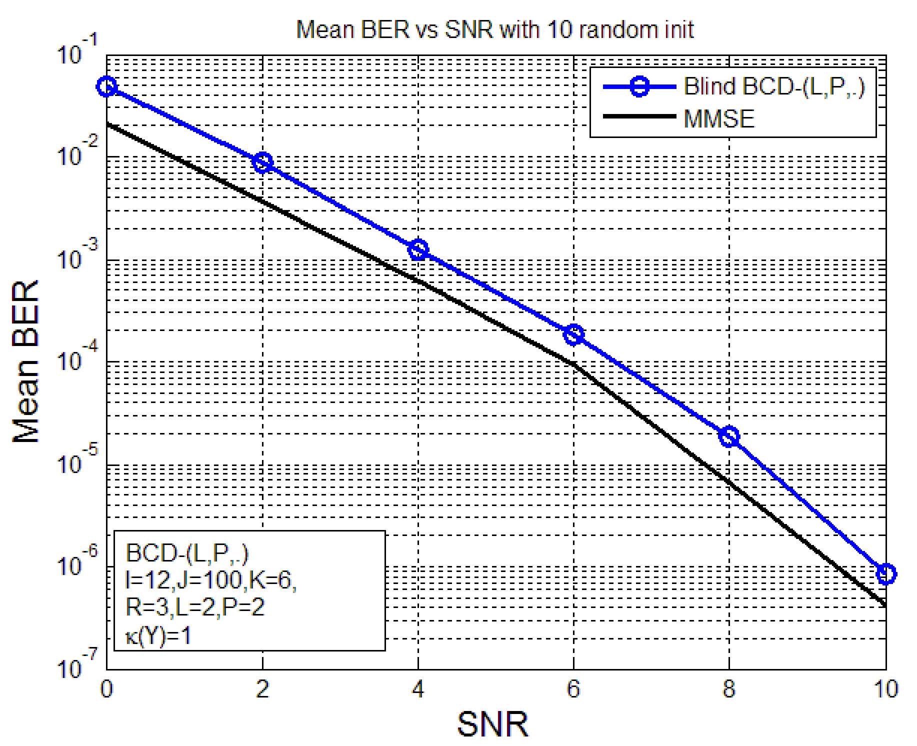
An application of the BCD-(L_r ,L_r ,1):

Blind Source Separation in telecommunications

I=12, J=100, L=2 for all users



K=4 antennas and R=5 users



K=6 antennas and R=3 users

Conclusion

- Block Component Decomposition in rank-(L_r ,L_r ,1) terms is a generalization of PARAFAC.

- Other BCD, even more general, have also been proposed [De Lathauwer & Nion]

- Algorithms: ALS coupled with Enhanced Line Search good compromise between complexity / convergence speed.

Algorithms based on Simultaneous Diagonalization (SD) also merits consideration (lower complexity than ALS and better accuracy)
→ on-going research

- Uniqueness: SD-based reformulation also yields relaxed uniqueness bound → on-going research

- Selection of the number of terms R and the rank L_r is important in practice (e.g. in telecoms R=number of users, L_r = user-dependent channel length)