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
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

\[ Y = U^{H} \times D \times V^{H} \]

\[ \begin{align*}
    Y & = \begin{bmatrix} J \end{bmatrix} \quad U \quad D \quad \begin{bmatrix} R \end{bmatrix} \\
    \end{align*} \]

\[ \begin{align*}
    Y & = U^{H} \times D \times V^{H} \\
    & = \begin{bmatrix} u_{1} \\
        \vdots \\
        u_{R} \end{bmatrix} + \ldots + \begin{bmatrix} v_{1}^{H} \\
        \vdots \\
        v_{R}^{H} \end{bmatrix}
\end{align*} \]

Tensor HOSVD (third-order case)

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

\[ Y = \mathcal{H} \times_{1} U \times_{2} V \times_{3} W \]

- One unitary matrix \((U, V, W)\) per mode
- \(\mathcal{H}\) is the representation of \(Y\) 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

\[ Y = U \Sigma V^H \]

\[ R = \begin{bmatrix} d_{11} & v_{11}^H \\ \vdots & \vdots \\ d_{RR} & v_{R1}^H \end{bmatrix} \]

\[ u_1 \quad + \ldots + \quad u_R \]

### PARAFAC decomposition

\[ Y = \sum_{r=1}^{R} \left( a_r b_r c_{r1} + \ldots + a_r b_r c_{rK} \right) \]

\[ Y_{1:} = \sum_{r=1}^{R} b_r c_{r1} \]

\[ Y_{:,k} = \sum_{r=1}^{R} a_r b_r c_{rk} = A \text{diag}(C(k,:)) B^T \]

\[ \Sigma \text{ is diagonal} \]

( if \( i=j=k, \ h_{ijk}=1 \), else, \( h_{ijk}=0 \) )
From PARAFAC/HOSVD to Block Components Decompositions (BCD) [De Lathauwer and Nion]

<table>
<thead>
<tr>
<th>BCD in rank ((L_r, L_r, 1)) terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{Y} ) = ( \mathbf{A}_1 \mathbf{B}_1^T ) + \ldots + ( \mathbf{A}_R \mathbf{B}_R^T )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BCD in rank ((L_r, M_r, .)) terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{Y} ) = ( \mathbf{A}_1 \mathbf{B}_1^T ) + \ldots + ( \mathbf{A}_R \mathbf{B}_R^T )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BCD in rank ((L_r, M_r, N_r)) terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{Y} ) = ( \mathbf{A}_1 \mathbf{B}_1^T ) + \ldots + ( \mathbf{A}_R \mathbf{B}_R^T )</td>
</tr>
</tbody>
</table>

\( \text{Y}_{IJK} = T_{1}^{B} A_{1}^{L} L_{r}^{1} + \ldots + T_{R}^{B} A_{R}^{L} L_{r}^{R} \)
Content of this talk

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

$$\mathcal{Y} = \frac{1}{\text{square}4}$$

$BCD - (L_r,L_r,1)$
**BCD - \((L_r, L_r, 1)\): Model ambiguities**

- **Unknown matrices:**
  
  \[
  A = \begin{bmatrix}
  A_1 & \ldots & A_R \\
  \end{bmatrix}
  \quad B = \begin{bmatrix}
  B_1 & \ldots & B_R \\
  \end{bmatrix}
  \quad C = \begin{bmatrix}
  \vdots \\
  c_1 & c_R \\
  \end{bmatrix}
  \]

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

  1. Arbitrary permutation of the R blocks in \(A\) and \(B\) and of the R columns of \(C\)
  2. Each block of \(A\) and \(B\) post-multiplied by arbitrary non-singular matrix, each column of \(C\) arbitrarily scaled.

  \[
  A = \begin{bmatrix}
  \end{bmatrix}
  \quad B = \begin{bmatrix}
  \end{bmatrix}
  \quad C = \begin{bmatrix}
  \end{bmatrix}
  \]

  \(A\) and \(B\) estimated up to multiplication by a **block-wise** permuted block-diagonal matrix and \(C\) by a permuted diagonal matrix.
Usual approach: estimate $A$, $B$ and $C$ by minimization of

$$\Phi = \left\| y - \sum_{r=1}^{R} ( A_r B_r^T ) \circ c_r \right\|_F^2$$

$\circ$ = outer product

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

Exploit algebraic structure of matrix unfoldings

- $y_{ijk} = Y_{ik}$
- $y_{ij} = Y_{ij}$
- $y_{i} = Y_{i}$
- $y_{j} = Y_{j}$

$$= Y_{I \times KJ}$$

$$= Y_{J \times IK}$$

$$= Y_{K \times JI}$$
\( Y_{K \times JI} = C \cdot Z_1(B, A) \)

\( Y_{J \times IK} = B \cdot Z_2(A, C) \rightarrow \Phi = \| Y_{K \times JI} - C \cdot Z_1(B, A) \|_F^2 \)

\( Y_{I \times KJ} = A \cdot Z_3(C, B) \)

\( \Phi = \| Y_{J \times IK} - B \cdot Z_2(A, C) \|_F^2 \)

\( \Phi = \| Y_{I \times KJ} - A \cdot Z_3(C, B) \|_F^2 \)

\( Z_1, Z_2 \) and \( Z_3 \) are built from 2 matrices only and have a block-wise Khatri-Rao product structure.

\[
\begin{align*}
\text{Initialisation}: & \quad \hat{A}^{(0)}, \hat{B}^{(0)}, k = 1 \\
\text{while} & \quad \left| \Phi^{(k-1)} - \Phi^{(k)} \right| > \epsilon \quad (\text{e.g. } \epsilon = 10^{-6}) \\
\hat{C}^{(k)} & = Y_{K \times JI} \cdot \left[ Z_1(\hat{B}^{(k-1)}, \hat{A}^{(k-1)}) \right]^+ \\
\hat{B}^{(k)} & = Y_{J \times IK} \cdot \left[ Z_2(\hat{A}^{(k-1)}, \hat{C}^{(k)}) \right]^+ \quad (1) \\
\hat{A}^{(k)} & = Y_{I \times KJ} \cdot \left[ Z_3(\hat{C}^{(k)}, \hat{B}^{(k)}) \right]^+ \quad (2) \\
k & \leftarrow k + 1
\end{align*}
\]
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 $\mathbf{Y} = \mathbf{Y}_1 + \ldots + \mathbf{Y}_R$ has a much higher norm than the R-1 others (e.g. « near-far » effect in telecommunications)

Observation:
ALS is fast in many problems, but sometimes, a long swamp is encountered before convergence.
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:
   \[ C^{(new)} = C^{(k-2)} + \rho \left( C^{(k-1)} - C^{(k-2)} \right) \]
   \[ B^{(new)} = B^{(k-2)} + \rho \left( B^{(k-1)} - B^{(k-2)} \right) \]
   \[ A^{(new)} = A^{(k-2)} + \rho \left( A^{(k-1)} - A^{(k-2)} \right) \]

2. Then ALS update
   \[ \hat{C}^{(k)} = Y_{K \times JI} \cdot \left[ Z_1 (\hat{B}^{(new)}, \hat{A}^{(new)}) \right]^{+} \] (1)
   \[ \hat{B}^{(k)} = Y_{J \times IK} \cdot \left[ Z_2 (\hat{A}^{(new)}, \hat{C}^{(k)}) \right]^{+} \] (2)
   \[ \hat{A}^{(k)} = Y_{I \times JK} \cdot \left[ Z_3 (\hat{C}^{(k)}, \hat{B}^{(k)}) \right]^{+} \] (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) »

\[
\Phi(A^{(\text{new})}, S^{(\text{new})}, H^{(\text{new})}) = \Phi(\rho) = 6^{\text{th}} \text{ order polynomial}.
\]

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

[Nion, De Lathauwer, 2006]

« Enhanced Line Search with Complex Step (ELSCS) »

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

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

Alternate update of $m$ and $\theta$:

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

- Update $\theta$: for $m$ fixed, $\frac{\partial \Phi(m, \theta)}{\partial \theta} = 6^{\text{th}} \text{ order polynomial in } t = \tan\left(\frac{\theta}{2}\right)$
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 \( y \)

STEP 2:
BCD of the small core tensor \( \mathcal{X} \) (compressed space)

STEP 3:
Come back to original space
+ a few refinement iterations in original space

- Compression \( \rightarrow \) 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 initialization, in same spirit as Direct Trilinear Decomposition (DTLD) used to initialize PARAFAC (not detailed in this talk).
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:
  \[
  M_n = WD_n W^T, \quad n=1,\ldots,R, \quad D_n \text{ is } R \times R \text{ diagonal}
  \]
- Advantage: Low complexity (only $R$ matrices of size $R \times R$ to diagonalize + direct use of existing fast algorithms designed for SD)
- SD reformulation yields a uniqueness bound generically more relaxed than Kruskal bound
  \[
  K \geq R \text{ et } \frac{I(I-1)}{2} \cdot \frac{J(J-1)}{2} \geq \frac{R(R-1)}{2}
  \]
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:

\[
M_n = WD_n W^T, \quad n=1,...,R, \quad D_n \text{ is } R \times R \text{ diagonal}
\]

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]:

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

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

\[
R \leq \min (IJ, K) \quad \text{and} \quad C_i^{L+1} \cdot C_j^{L+1} \geq C_{R+L}^{L+1} - R \quad (2)
\]

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

---

New Bound much more relaxed.

---

<table>
<thead>
<tr>
<th>L</th>
<th>R max</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>180</td>
</tr>
<tr>
<td>4</td>
<td>160</td>
</tr>
</tbody>
</table>

---
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).

→ Important practical considerations:
  - 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)
**Problem:** Given a tensor $\otimes_J$, how to estimate the number of terms $R$ and the rank $L_r$ of the matrices $A_r$ and $B_r$ that yield a reasonable ($L_r, L_r, 1$) model?

\[
Y_{IJK} = A_1^{L_1} B_1^{T} + \ldots + A_R^{L_R} B_R^{T}
\]

- **Criterion 1:** Simple approach: examine 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 $\min(K,J) \geq N$
  - $Y_{KJxI}$ 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.

\[ Y_{IKJ} = A_{IR} \cdot C_{KR} \cdot B^T_{RJ} \]

The core tensor is diagonal:

\[ H_{ijk} = \begin{cases} 1 & \text{if } i = j = k \\ 0 & \text{otherwise} \end{cases} \]

**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 \( \hat{H} \), and measure the diagonality of the core tensor:

\[ C = 100 \left( 1 - \frac{\| H - \hat{H} \|^2_F}{R} \right) \]

- Examine the core consistency measurement to select an optimal \( R \)
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 ».

\[
Y = \sum_{r=1}^{R} c_r T^T \mathbf{B}_r^T \mathbf{A}_r^T + \ldots + c_1 T^T \mathbf{B}_1^T \mathbf{A}_1^T
\]

\[
N = \sum_{r=1}^{R} L_r
\]
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 $\mathcal{Y}$.
- and measure the block - diagonality of the core tensor:

\[
C_{COR} = 100 \left( 1 - \frac{\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2}{RL} \right)
\]

- Examine 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:

\[
C_{Fit} = 100 \left( 1 - \frac{\|y - \hat{y}\|_F^2}{\|y\|_F^2} \right)
\]
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: \( R_{\text{try}} = \{1,2,3,4,5,6\} \) and \( L_{\text{try}}=\{1,2,3,4\} \)

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

\( C_{\text{Fit}} = \)

\[
\begin{array}{cccc}
22.3 & 36.6 & 38.1 & 39.3 \\
38.6 & 66.6 & 67.8 & 69.1 \\
56.3 & \textbf{91.2} & 91.3 & 91.4 \\
71.4 & 91.5 & 91.7 & 91.8 \\
84.1 & 91.7 & 91.9 & 92.1 \\
91.5 & 92.0 & 92.3 & 92.4 \\
\end{array}
\]

\( C_{\text{COR}} = \)

\[
\begin{array}{cccc}
100 & 100 & 100 & 100 \\
99 & 99.8 & <0 & <0 \\
98.9 & \textbf{99.4} & <0 & <0 \\
84.8 & 30.9 & <0 & <0 \\
<0 & <0 & <0 & <0 \\
<0 & <0 & <0 & <0 \\
\end{array}
\]

\( \rightarrow \) L=2 and R=3 corresponds to the intersection of the acceptable values of Fit and the ones for Core Consistency.
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: \( R_{\text{try}} = \{1, 2, 3, 4, 5, 6\} \) and \( L_{\text{try}} = \{1, 2, 3, 4, 5\} \)

\[
\begin{array}{cccccc}
20.3 & 32.8 & 38.1 & 40.4 & 41.6 \\
37.8 & 60.8 & 68.4 & 69.8 & 70.4 \\
54.2 & \textbf{81.3} & \textbf{91.4} & 91.4 & 91.5 \\
68.7 & 88.1 & 91.7 & 91.8 & 91.9 \\
78.1 & 91.4 & 91.9 & 91.1 & 92.2 \\
82.8 & 91.9 & 92.3 & 92.5 & 92.6 \\
\end{array}
\]

\[
\begin{array}{cccccc}
100 & 100 & 100 & 100 & 100 \\
95.2 & 96.1 & 55.1 & < 0 & < 0 \\
94.1 & \textbf{64.2} & \textbf{59.9} & < 0 & < 0 \\
60.3 & < 0 & < 0 & < 0 & < 0 \\
< 0 & < 0 & < 0 & < 0 & < 0 \\
< 0 & < 0 & < 0 & < 0 & < 0 \\
\end{array}
\]

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

\( \rightarrow \text{Find with other criteria to help in the final decision} \)
Criterion 4: use the BCD-(L,L,1) structure

\[ Y = \sum_{r=1}^{R} A_r B_r^T + \ldots + A_1 B_1^T \]

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

Repetition of the vectors \( c_r \) 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
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.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.15</th>
<th>0.99</th>
<th>0.09</th>
<th>0.14</th>
<th>0.86</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>1</td>
<td>0.15</td>
<td>0.39</td>
<td>0.95</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.15</td>
<td>1</td>
<td>0.10</td>
<td>0.13</td>
<td>0.86</td>
<td></td>
</tr>
<tr>
<td>0.09</td>
<td>0.39</td>
<td>0.10</td>
<td>1</td>
<td>0.24</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>0.13</td>
<td>0.95</td>
<td>0.13</td>
<td>0.24</td>
<td>1</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>0.86</td>
<td>0.41</td>
<td>0.86</td>
<td>0.12</td>
<td>0.45</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0.17</th>
<th>0.17</th>
<th>0.18</th>
<th>0.11</th>
<th>0.09</th>
<th>0.11</th>
<th>0.99</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>0.10</td>
<td>0.12</td>
<td>0.10</td>
<td>0.17</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>0.17</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>0.10</td>
<td>0.11</td>
<td>0.10</td>
<td>0.17</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>0.18</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
<td>0.13</td>
<td>0.14</td>
<td>0.13</td>
<td>0.18</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
<td>0.13</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>0.12</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>0.09</td>
<td>0.12</td>
<td>0.11</td>
<td>0.14</td>
<td>0.99</td>
<td>1</td>
<td>0.99</td>
<td>0.10</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>0.11</td>
<td>0.10</td>
<td>0.10</td>
<td>0.13</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
<td>0.12</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.17</td>
<td>0.17</td>
<td>0.18</td>
<td>0.12</td>
<td>0.10</td>
<td>0.12</td>
<td>1</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.18</td>
<td>0.18</td>
<td>0.19</td>
<td>0.13</td>
<td>0.11</td>
<td>0.13</td>
<td>0.99</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Clustering in \(R=3\) groups of 3 vectors « good »
Applications

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 $\text{BCD-}(L_r,L_r,1)$: 

**Blind Source Separation in telecommunications**

Decompose $\mathbf{y}$ 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.]

\[ y_i = a_1 s_1 + \ldots + a_R s_R \]

- \(I\) = 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]

\[
Y_{K \times J} = \sum_{r=1}^{R} H_{r} L_{r} S_{r}^{T} \]

\( H_{r} \rightarrow \) Channel matrix (channel impulse response convolved with CDMA code)
\( S_{r} \rightarrow \) Symbol matrix, holds the J symbols of interest for user r
\( a_{r} \rightarrow \) 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 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)