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
I. Introduction

What is a tensor?

Tensor of order $N =$ Array with $N$ dimensions

For $N>2$, « Higher-Order Tensors »

\[ y \]  
= 1st-order tensor

\[ \mathbf{Y} \]  
= 2nd-order tensor

\[ \mathbf{Y} \]  
= 3rd-order tensor
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)
I. Introduction

Multi-Way Data

A 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
I. Introduction

**Tensor Unfolding ("matricization")**

\[
\begin{align*}
Y & = Y_{I \times K \times J} \\
Y_k & = Y_{I \times J} \\
Y_i & = Y_{J \times I} \\
Y_j & = Y_{K \times I}
\end{align*}
\]

**Multi-Way Analysis?**

- One can choose one matrix representation of \( Y \) 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)

\[ Y = U S V^H \]

- \( U^H U = I \) and \( V^H V = I \) \( \rightarrow \) unitary matrices
- \( S = \text{diag}(\sigma_1, \ldots, \sigma_R) \) \( \rightarrow \) Singular values in decreasing order

If \( \text{rank}(Y) > R \), this truncated SVD is the best rank-\( R \) approx. of \( Y \)

In general a matrix factorization \( Y = UV^H \) is \textit{not} unique:

\[ Y = UV^H = UPP^{-1}V^H \]

The SVD is unique because of unitary constraints on \( U \) and \( V \) and ordering constraint of the singular values in \( S \)
I. Tensor Decompositions

Tucker-3 Decomposition [Tucker 1966]

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

\[ y = \mathcal{H} \times_1 A \times_2 B \times_3 C \]

- 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 \(y\) 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).
I. Tensor Decompositions

Uniqueness of Tucker-3 Decomposition

- 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]

\[ Y_1 = \text{truncated Matrix SVD of } Y \]

\[ Y_1 = U S V^H \]

\[ Y_1 \] is the best lower rank approximation of \( Y \) (in the Frobenius norm sense):

\[ \text{Min } ||Y-Y_1||_F \]

s.t. \( Y_1 \) is rank-R

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

\[ \text{Min } \left\| \mathcal{Y} - \mathcal{A} \mathcal{B}' \mathcal{C} \right\|_F \]

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 \( \mathcal{A}, \mathcal{B} \) and \( \mathcal{C} \).
I. Tensor Decompositions

PARAFAC Decomposition [Harshman 1970]

\[ Y = \sum_{r=1}^{R} a_r b_r c_r = \sum_{r=1}^{R} \mathbf{a}_r \mathbf{b}_r \mathbf{c}_r \]

Where:
- \( a_r \) is a vector
- \( b_r \) is a vector
- \( c_r \) is a vector
- \( \mathbf{a}_r \) is a matrix
- \( \mathbf{b}_r \) is a matrix
- \( \mathbf{c}_r \) is a matrix

\( \mathbf{C} \) is diagonal
- \( \mathbf{C}_{ij} = 1 \) if \( i=j=k \)
- \( \mathbf{C}_{ij} = 0 \) otherwise

\( \mathbf{Y} \) is set of \( K \) matrices of the form:

\[ \mathbf{Y}(i;,j,k) = \mathbf{A} \text{diag}(\mathbf{C}(k;)) \mathbf{B}^T \]
Under mild conditions (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)
I. Tensor Decompositions

Uniqueness of PARAFAC Decomposition (2)

Uniqueness condition [Kruskal, 1977]

\[ k_A + k_B + k_C \geq 2R + 2 \]  
(1)

\( k_A \) is the Kruskal-rank of \( A \)

Generically, \( k_A = \min(I,R) \)

\[ \min(I,R) + \min(J,R) + \min(K,R) \geq 2(R+1) \]  
(2)

Relating on (real and complex cases) on ( ) so on ( )

[De Lathauwer 2005]:

\[ J \geq R \text{ et } \frac{I(I-1) + K(K-1)}{2} \geq \frac{R(R-1)}{2} \]  
(3)
I. Tensor Decompositions

**PARAFAC vs Tucker 3**

**PARAFAC**

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

\[ \mathcal{H} \text{ is diagonal} \]

\[ L=M=N \rightarrow A, B \text{ and } C \text{ have the same nb. of columns} \]

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

**TUCKER 3**

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

\[ \mathcal{H} \text{ is not diagonal} \]

\[ L\neq M\neq N \rightarrow A, B \text{ and } C \text{ do not necessarily have the same nb. of columns} \]

Not unique:
Rotational freedom still remains.
I. Tensor Decompositions

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

\[
Y = A_1^T B_1 + \ldots + A_R^T B_R
\]

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

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

- BCD-(\(L_r, L_r, 1\)) is said unique if the only remaining ambiguities are:
  \(\rightarrow\) Arbitrary permutation of the blocks in \(A\) and \(B\) and of the columns of \(C\)
  \(\rightarrow\) Rotational freedom of each block (block-wise subspace estimation) + scaling ambiguity on the columns of \(C\)
## I. Tensor Decompositions

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

<table>
<thead>
<tr>
<th>Sufficient bound 1</th>
<th>$LR \leq IJ$ and $\min\left(\frac{I}{L},R\right)+\min\left(\frac{J}{L},R\right)+\min(K,R) \geq 2(R+1)$ (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[De Lathauwer SIMAX 2008]</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sufficient bound 2</th>
<th>$R \leq \min(IJ,K)$ and $C_i^{L+1} \cdot C_j^{L+1} \geq C_{R+L}^{L+1} - R$ (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[Nion, PhD Thesis, 2007]</strong></td>
<td></td>
</tr>
</tbody>
</table>

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

---

**Diagram:**

- **bom 1, L=2**
- **bom 2, L=2**
- **bom 1, L=3**
- **bom 2, L=3**
- **bom 1, L=4**
- **bom 2, L=4**

- **R max**
- **I=J**
I. Tensor Decompositions

**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:**
  
  \[
  \begin{align*}
  A &= \begin{pmatrix} A_1 & \cdots & A_R \end{pmatrix} & B &= \begin{pmatrix} B_1 & \cdots & B_R \end{pmatrix} & C &= \begin{pmatrix} C_1 & \cdots & C_R \end{pmatrix} \\
  \end{align*}
  \]
  
  \[
  \begin{pmatrix} L_1 & L_R \end{pmatrix} & \begin{pmatrix} M_1 & M_R \end{pmatrix} & \begin{pmatrix} N_1 & N_R \end{pmatrix}
  \]
  
  \[
  \mathcal{H} = \begin{pmatrix} \mathcal{H}_1 & \cdots & \mathcal{H}_R \end{pmatrix}
  \]

- **Ambiguities:** same as Tucker model for each of the \(R\) components
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

\[
\Phi = \left\| \mathcal{Y} - Tens(\hat{H}, \hat{S}, \hat{A}) \right\|_F^2 \quad \text{Tens = PARAFAC or BCD-(L,L,1) or BCD-(L,P,..)}
\]

**Main idea:** exploit the structure of the three matrix unfoldings simultaneously

\[
\begin{align*}
Y_{K \times J_1} &= C \cdot Z_1(B, A) \\
Y_{J_1 \times I_1} &= B \cdot Z_2(A, C) \\
Y_{I_1 \times K_1} &= A \cdot Z_3(C, B)
\end{align*}
\]

\[
\begin{align*}
\Phi &= \left\| Y_{K \times J_1} - C \cdot Z_1(B, A) \right\|_F^2 \\
\Phi &= \left\| Y_{J_1 \times I_1} - B \cdot Z_2(A, C) \right\|_F^2 \\
\Phi &= \left\| Y_{I_1 \times K_1} - A \cdot Z_3(C, B) \right\|_F^2
\end{align*}
\]

\( Z_1, Z_2 \) 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 \( A = [A_1, \ldots, A_R], B = [B_1, \ldots, B_R] \) and \( C = [C_1, \ldots, C_R] \) in the Least Squares sense.

- Each update = minimization of the cost function w.r.t. one the 3 matrix unfoldings

---

**Initialization:** \( \hat{A}^{(0)}, \hat{B}^{(0)}, k = 1 \)

\[
\text{while } |\Phi^{(k-1)} - \Phi^{(k)}| > \epsilon \quad (\text{e.g. } \epsilon = 10^{-6})
\]

1. \( \hat{C}^{(k)} = Y_{K \times J_l} \cdot [Z_1(\hat{B}^{(k-1)}, \hat{A}^{(k-1)})]^\dagger \)

2. \( \hat{B}^{(k)} = Y_{J \times I_K} \cdot [Z_2(\hat{A}^{(k-1)}, \hat{C}^{(k)})]^\dagger \)

3. \( \hat{A}^{(k)} = Y_{I \times K_J} \cdot [Z_3(\hat{C}^{(k)}, \hat{B}^{(k)})]^\dagger \)

\( k \leftarrow k + 1 \)
ALS algorithm: problem of swamps

Long Swamps typically occur when:
- The loading matrices of the decomposition (i.e. the objective matrices) are ill-conditioned
- The updated matrices become ill-conditioned (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.

27000 iterations!
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:
   \[
   \begin{align*}
   C^{(new)} &= C^{(k-2)} + \rho \left( C^{(k-1)} - C^{(k-2)} \right) \\
   B^{(new)} &= B^{(k-2)} + \rho \left( B^{(k-1)} - B^{(k-2)} \right) \\
   A^{(new)} &= A^{(k-2)} + \rho \left( A^{(k-1)} - A^{(k-2)} \right)
   \end{align*}
   \]

2. Then ALS update
   \[
   \begin{align*}
   \hat{C}^{(k)} &= Y_{K\times JI} \cdot \left[ Z_1(\hat{B}^{(new)}, \hat{A}^{(new)}) \right]^+ \\
   \hat{B}^{(k)} &= Y_{J\times IK} \cdot \left[ Z_2(\hat{A}^{(new)}, \hat{C}^{(k)}) \right]^+ \\
   \hat{A}^{(k)} &= Y_{I\times KJ} \cdot \left[ Z_3(\hat{C}^{(k)}, \hat{B}^{(k)}) \right]^+
   \end{align*}
   \]

Choice of \( \rho \) crucial
\( \rho = 1 \) annihilates LS step (i.e. we get standard ALS)
**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 $\Phi(A^{(new)}, S^{(new)}, H^{(new)}) = \Phi(\rho) = 6^{th}$ order polynomial.

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

[Nion, De Lathauwer, 2006]

« Enhanced Line Search with Complex Step (ELSCS) »

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

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

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

- Update $m$ : for $\theta$ fixed, $\frac{\partial \Phi(m, \theta)}{\partial m} = 5^{th}$ order polynomial in $m$
- Update $\theta$ : for $m$ fixed, $\frac{\partial \Phi(m, \theta)}{\partial \theta} = 6^{th}$ order polynomial in $t = \tan\left(\frac{\theta}{2}\right)$
Improvement 1 of ALS: Line Search

«easy» problem

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

«difficult» problem

2000 iterations

27000 iterations
Improvement 2 of ALS: Compression

**STEP 1:**
Fit a Tucker Model on $\mathcal{X}$

**STEP 2:**
Fit the model on the small core tensor $\mathcal{X}$ (compressed space)

**STEP 3:**
Come back to 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?
Improvement 3 of ALS: Good initialization

Slices $Y_k$ (IxJ) of $y$: 
\[
\begin{align*}
Y_1 &= H \cdot \Lambda_1 \cdot S^T \\
Y_2 &= H \cdot \Lambda_2 \cdot S^T \\
& \vdots \\
Y_k &= H \cdot \Lambda_k \cdot S^T
\end{align*}
\]
where the $\Lambda_i$ are diagonal.

For PARAFAC: if $R \leq \min(I, J)$, the slices $Y_k$ are generically rank-R.

For any pair $(k_1, k_2)$:
\[
Y_{k_1} \cdot (Y_{k_2})^\dagger = H \cdot (\Lambda_{k_1} \cdot \Lambda_{k_2}^{-1}) \cdot H^\dagger
\]

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

→ 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)

→ Important practical considerations:
  - 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
Applications

**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
Applications

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

Standard approach: 2-Way PCA

1260 (28x3x5x3) pixels

\[ \begin{pmatrix} Y \\ \Sigma_1 \\ U_{\text{people}} \end{pmatrix} = \text{SVD} \]

\[ U_{\text{pixel}} (7943 \times 1260) \]
spans the space of images

\[ \rightarrow \text{1 image represented by one vector of 1260 coefficients in } V \]

\[ \rightarrow \text{1 person represented by a set of 45 vectors in } V \]

**Input Image** \( d \) (7943x1)

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

2) \( \min_i ||c - v_i|| \) to associate score vector \( c \) to one person
**Application 1: Tensor Faces & Face Recognition**

[Vasilescu & Terzopoulos, 2003]

1260 (28x3x5x3) → N-Way PCA

7943 pixels → tensor \( \mathcal{Y} \) (7943x5x3x3x28)

\[ \mathcal{Y} = \mathcal{K} \times_1 U_{\text{pixels}} \times_2 U_{\text{views}} \times_3 U_{\text{illums}} \times_4 U_{\text{express}} \times_5 U_{\text{people}} \]

- \( U_{\text{pixels}} \) (7943x7943) spans the space of images
- \( U_{\text{views}} \) (5x5) spans the space of viewpoint parameters
- \( U_{\text{illums}} \) (3x3) spans the space of illumination parameters
- \( U_{\text{express}} \) (3x3) spans the space of expression parameters
- \( U_{\text{people}} \) (28x28) spans the space of people parameters

\( \mathcal{K} \) 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]

1) For all triplets (view,illums,express), build the basis $B_{v,i,e}$ (7943x28) and project unknown image $c = B_{v,i,e} d$

2) Compare the 28x1 score vector $c$ to the loadings in $U_{people}$

$$\min_i ||c-u_i||$$

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]

<table>
<thead>
<tr>
<th>Data set:</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ 2 chemical samples, each containing different and unknown concentrations of 3 unknown chemical components.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Goal:</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ Find which chemical components are present in the samples</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method: fluorescence</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ Excitation of the samples with 51 wavelengths (250-300nm)</td>
</tr>
<tr>
<td>→ Measure of the intensity of emission over 201 wavelengths (250-450nm)</td>
</tr>
</tbody>
</table>
Applications

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

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

Fit PARAFAC model with $R=3$ components

Identification of 3 chemical components with only 2 samples

→ thanks to uniqueness of PARAFAC decomposition
Applications


Estimated emission spectrum

True excitation spectrum

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

\[ \mathbf{y} = \mathbf{a}_1 \mathbf{c}_1 + \ldots + \mathbf{a}_R \mathbf{c}_R \]

- \( I \) = 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.

[Sidiropoulos et al., 2001]
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]

\[ H_r \rightarrow \text{Channel matrix (channel impulse response convolved with CDMA code)} \]
\[ S_r \rightarrow \text{Symbol matrix, holds the J symbols of interest for user r} \]
\[ a_r \rightarrow \text{Response of the K antennas to the angle of arrival (steering vector)} \]
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]

\[ \mathbf{Y} = \sum_{r=1}^{R} \mathbf{H}_r \]

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

**\( \mathbf{S}_r \) → Symbol matrix, holds the J symbols of interest for user r**

**\( \mathbf{A}_r \) → Response of the K antennas to the angles of arrival (steering vectors)**
Applications

**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)**

**Goal:** estimate the $I$ unknown sources $s_1, \ldots, s_I$, from the $J$ recordings $m_1, \ldots, m_J$ only. (« blind source separation (BSS) »)
**Application 4:**

Blind Source Separation (instantaneous mixtures)

Data Model for linear instantaneous mixtures:

![Diagram showing the data model with N samples, J signals, observed matrix Y, mixing matrix H, and source matrix S.]

**Issues:**

→ 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)?
Matrix factorization not unique:

\[
\begin{bmatrix}
N \\
J
\end{bmatrix}
Y
= 
\begin{bmatrix}
J \\
I
\end{bmatrix}
H
P
P^{-1}
\begin{bmatrix}
N \\
I
\end{bmatrix}
S
\]

The SVD of \( Y \) would give us the subspaces that generate \( H \) and \( S \), but not \( H \) and \( S \) themselves \( \rightarrow \text{We need more assumptions!} \)

Assumption: The I sources are statistically independent

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

\[\longrightarrow\text{Find } H \text{ that makes the source estimates as much independent as possible.}\]

\[\longrightarrow\text{Use of Second-Order or Higher-Order Statistics (SOS or HOS)}\]

+ Application-specific assumptions to reduce the ambiguity:

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

**Application 4:**

Blind Source Separation (instantaneous mixtures)

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

\[
C_k = E[y_t y_{t-\tau_k}^H] = HE[s_t s_{t-\tau_k}^H]H^H = HD_k H^H
\]

K delays \(\rightarrow\) K covariance matrices

\[
\begin{align*}
C_1 &= HD_1 H^H \\
\vdots &= \vdots \\
C_K &= HD_K H^H
\end{align*}
\]

Use existing algorithms for **Joint Diagonalization** of a set of matrices to find \(H\)

SOBI relies on simultaneous diagonalization algorithms \(\rightarrow\) does not work in under-determined cases (i.e., when \(H\) is fat)
Blind Source Separation (instantaneous mixtures)

« Second-Order-Blind-Identification of Under-determined mixtures » (SOBIUM)
[Castaing & De Lathauwer 2006]

\[
\begin{align*}
C_1 &= HD_1H^H \\
\vdots &= \vdots \\
C_K &= HD_KH^H
\end{align*}
\]

\(C = JDH^H\) 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):

<table>
<thead>
<tr>
<th>J</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_{\text{max}})</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>26</td>
</tr>
</tbody>
</table>
**Application 5:**

**Blind Source Separation (convolutive mixtures)**

\[ Y = HS \rightarrow \text{instantaneous mixtures} \]

Multiple reverberations on the walls \(\rightarrow\) separation of convolutive mixture

\[
y(t) = H \ast s(t) = \sum_{l=0}^{L-1} H(l) \ s(t - l)
\]

**DFT**

\[
y(f, t) = H(f) \ s(f, t), \quad f = 1, \ldots, F
\]

Solve one instantaneous ICA problem for each frequency \(\rightarrow\) apply existing ICA techniques for instantaneous mixtures
Application 5:

Blind Source Separation (convolutive mixtures)

\[ y(f, t) = H(f) s(f, t), \]
\[ f = 1, ..., F \]

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:

\[ s(f, t) = H^\dagger(f)y(f, t) \]

\[ \rightarrow \text{arbitrary scaling and permutation of columns of } H(f) \text{ at each frequency} \]

\[ \rightarrow \text{Under-determined cases: we can not compute } s(f, t) = H^\dagger(f)y(f, t) \]
Application 5:

Blind Source Separation (convolutive mixtures)

« PARAFAC-Based Separation of convolutive speech mixtures »
[Nion, Mokios, Sidiropoulos & Potamianos 2008]

Applications

Example 1:
I=4 speech signals,
J=8 microphones

\[ \hat{S}_1 \quad \hat{S}_2 \quad \hat{S}_3 \quad \hat{S}_4 \]

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

\[ \hat{S}_1 \hat{S}_2 \hat{S}_3 \]

Room Impulse Response (T\(_{60}\)=200 ms)
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.
Applications

Application 6:

Target localization in MIMO radars

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

\[ Y_q = B(\theta_r) \sum_q A^T(\theta_t) + Z_q, \quad q = 1, \ldots, Q \]

\[ \begin{array}{cccc}
M_t \times M_t & M_r \times K & K \times K & K \times M_t \\
\text{diagonal} & & & \text{AWGN}
\end{array} \]

\text{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
Applications

Application 6:

Target localization in MIMO radars

\[ Y_q = B(\theta_r) \Sigma_q A^T(\theta_t) + Z_q, \quad q = 1, \ldots, Q \]

« 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
Applications

Application 6:

Target localization in MIMO radars

« Beamforming-based approach »:

\[ P(\theta_t, \theta_r) \]

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.
Applications

Application 6:

Target localization in MIMO radars

PARAFAC vs Capon

- Q=200 pulses
- Target 1: (-10°, -20°)
- Target 2: (-14°, -24°)

Graph showing the comparison between PARAFAC and Capon methods with different antenna configurations (M_t=M_r=2, 6, 8) and varying SNR (dB).
Application 7:

Tracking the PARAFAC decomposition

« Adaptive algorithms to track the PARAFAC decomposition »

[Nion & Sidiropoulos 2008]

Time

New Slice

LINK = ADAPTIVE ALGORITHMS
Applications

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
Applications

Application 7:

Tracking the PARAFAC decomposition

« Adaptive algorithms to track the PARAFAC decomposition »

[Nion & Sidiropoulos 2008]

Example 2: BSS

![Graph showing SIR (signal-to-interference ratio) over duration of recording.]

- S1 is instantaneously moved from 90° to 135° while S2 is fixed.
- Seq. 1: Initialization
- Seq. 2: PARAFAC-RLST S1 at 90°, S2 at 0°
- Seq. 3: PARAFAC-RLST S1 at 135°, S2 at 0°
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 $\rightarrow$ 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.