# Additional structure in the JADE algorithm for Independent Component Analysis

Pieter Penninckx, Lieven De Lathauwer

Monopoli, 2010

# Aim

To improve the accuracy

of the JADE algorithm

for applications in Independent Component Analysis

- Problem formulation

  Independent Component Analysis
- The JADE algorithm
- The JADE+ algorithm
- Experimental comparison

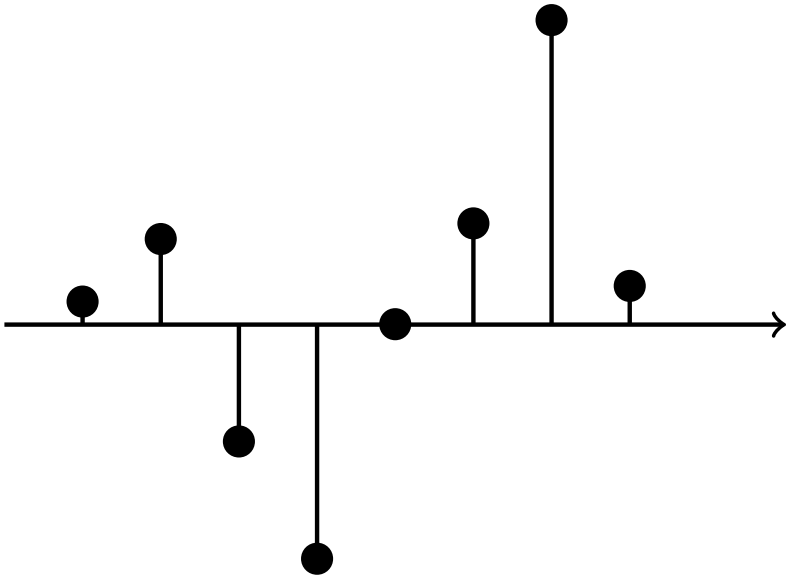
# Independent component analysis

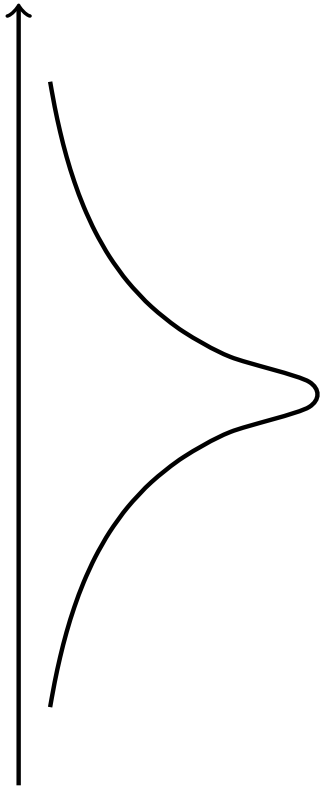
## Aim


| | |
|---|---|
| Given | sensor signals |
| Asked | source signals |
| Indeterminacy | number of signals, order, scaling |

# Independent component analysis

## Signals as random numbers





# Independent component analysis

## Data model

$J$ sources


$I$ sensors

| | | |
|---|---|---|
| sources | $S \in \mathbb{R}^J$ | random vector |
| | non-Gaussian, statistically independent | |
| sensors | $X \in \mathbb{R}^I$ | random vector |
| mixing | $M \in \mathbb{R}^{I \times J}$ | matrix of full column rank |

$$X = MS$$

# Cumulant tensors are higher order statistics

$Y \in \mathbb{R}^N$ random vector

## Cumulant tensors

| | | |
|---|---|---|
| first order | mean | $\mathbb{R}^N$ |
| second order | covariance matrix | $\mathbb{R}^{N \times N}$ |
| third order | $\mathcal{C}_Y^{(3)}$ | $\mathbb{R}^{N \times N \times N}$ |
| fourth order | $\mathcal{C}_Y^{(4)}$ | $\mathbb{R}^{N \times N \times N \times N}$ |

# Cumulant tensors are supersymmetric

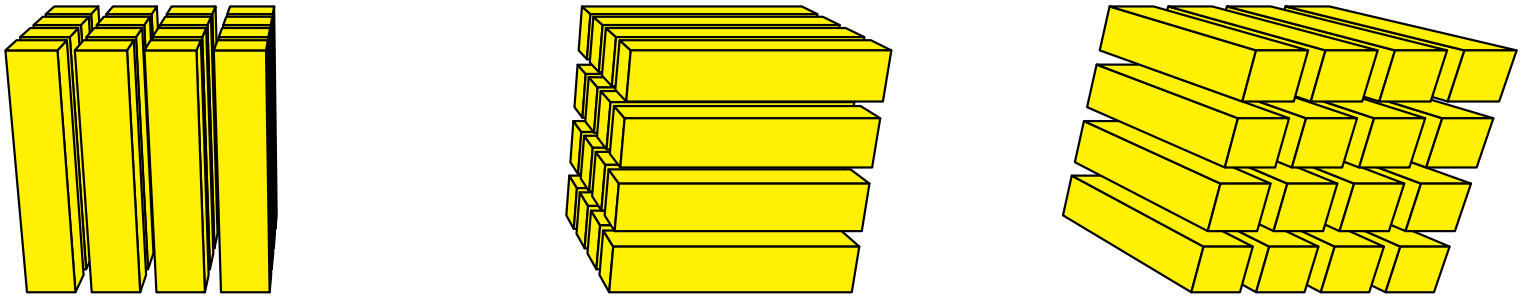
$$(\mathcal{C}_Y)_{i_1,\ldots,i_k} = (\mathcal{C}_Y)_{\sigma(i_1,\ldots,i_k)}$$

for any permutation $\sigma$

# The mode-$k$ product of a tensor with a matrix

## Example

$$(\mathcal{T} \times_3 A)_{i,j,k,l} = \sum_m \mathcal{T}_{i,j,m,l} A_{m,k}$$



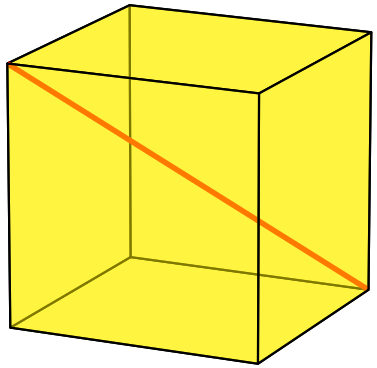
# Multilinearity of cumulant tensors

$$X = MS$$

$$\mathcal{C}_X^{(K)} = \mathcal{C}_{MS}^{(K)} = \mathcal{C}_S^{(K)} \times_1 M \cdots \times_K M$$

# Superdiagonal cumulant tensors

$S = (s_1, \ldots, s_J)$
Because $s_1, \ldots, s_J$ are statistically independent,
$\mathcal{C}_S$ is a superdiagonal tensor:



# How to compute the mixing matrix $M$

$$X = MS \iff \mathcal{C}_X^{(4)} = \mathcal{C}_S^{(4)} \times_1 M \times_2 M \times_3 M \times_4 M$$

# How to compute the mixing matrix $M$

pseudo-inverse of

$$X = MS \iff \mathcal{C}_X^{(4)} = \mathcal{C}_S^{(4)} \times_1 M \times_2 M \times_3 M \times_4 M$$

$$S = M^\dagger X \iff \mathcal{C}_S^{(4)} = \underbrace{\mathcal{C}_X^{(4)} \times_1 M^\dagger \times_2 M^\dagger \times_3 M^\dagger \times_4 M^\dagger}_{\text{minimize 'non-superdiagonal'}}$$

# How to compute the mixing matrix $M$

$$\mathcal{C}_S^{(4)} = \mathcal{C}_X^{(4)} \times_1 M^\dagger \times_2 M^\dagger \times_3 M^\dagger \times_4 M^\dagger$$

$M^\dagger = U \Delta V \qquad$ (SVD)

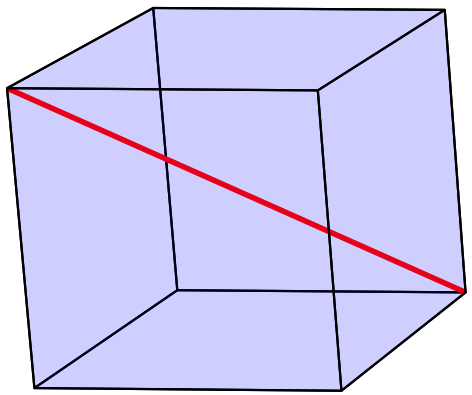
$\Delta V$ can be computed from second order statistics.

$Z := \Delta V X$

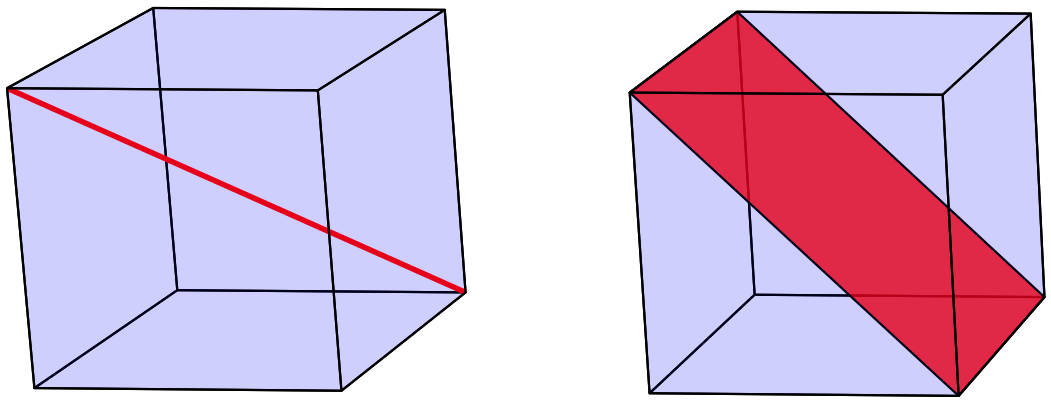
$UZ = U \Delta V X = M^\dagger X = S$

$$\mathcal{C}_S^{(4)} = \mathcal{C}_{UZ}^{(4)} = \mathcal{C}_Z^{(4)} \times_1 U \times_2 U \times_3 U \times_4 U$$

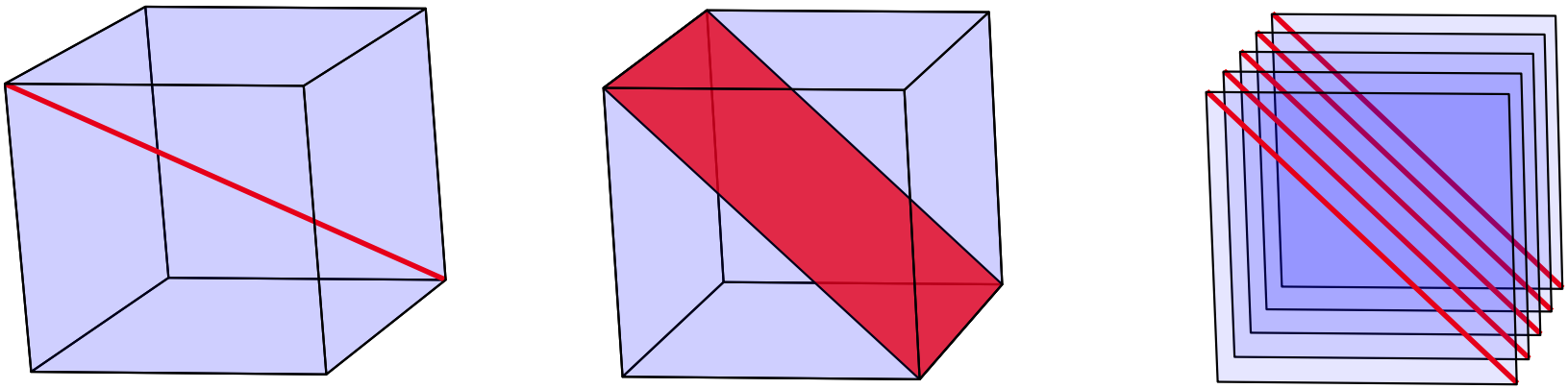
# The JADE algorithm



# The JADE algorithm



# The JADE algorithm



# The core step of JADE is a Jacobi rotation

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \quad \begin{pmatrix} * & * \\ * & * \end{pmatrix} \quad \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

$c = \cos(\theta), s = \sin(\theta)$

$(\cos 2\theta, \sin 2\theta)$: dominant eigenvector of a matrix

# The core step of JADE is a Jacobi rotation

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & -s \\ 0 & 0 & 1 & 0 \\ 0 & s & 0 & c \end{pmatrix} \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{pmatrix}$$

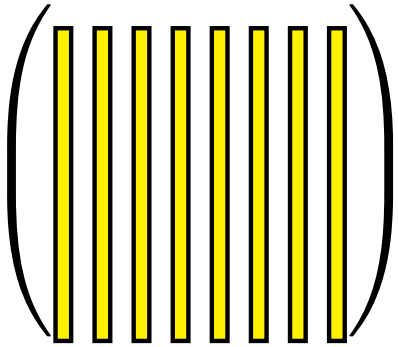
$c = \cos(\theta), s = \sin(\theta)$

$(\cos 2\theta, \sin 2\theta)$: dominant eigenvector of a matrix

sweep: iterate over all principal $2 \times 2$ submatrices

# Hidden additional structure

The $J^2$ matrix slices of $\mathcal{C}_Z^{(4)} \in \mathbb{R}^{J \times J \times J \times J}$ are linear combinations of a set of $J$ matrices.

$C^Z =$  $= \begin{pmatrix} E & 0 \end{pmatrix} \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} E & 0 \end{pmatrix}^T$

$$= E \Lambda E^T$$

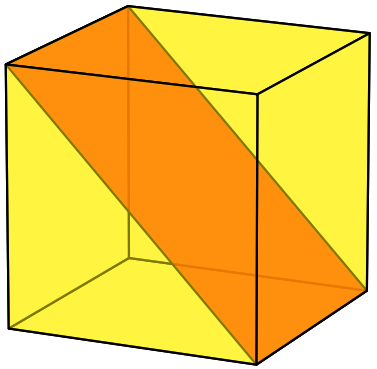
$$= E \Lambda^{1/2} \Lambda^{1/2} E^T$$

# Hidden additional structure

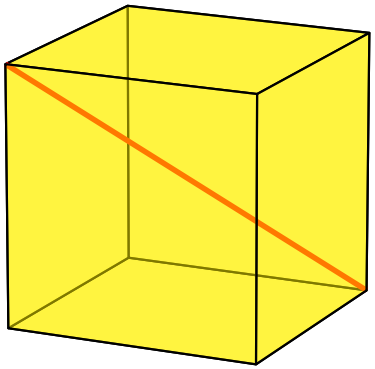
$$\mathcal{T} = \Delta \times_1 U \times_2 U \times_3 W$$

$\Delta$ superdiagonal tensor, $U, W$ orthogonal

$$\mathcal{T} \times_1 U^{-1} \times_2 U^{-1} = \Delta \times_3 W =$$



$$\mathcal{T} \times_1 U^{-1} \times_2 U^{-1} \times_3 W^{-1} = \Delta =$$



# The JADE+ algorithm

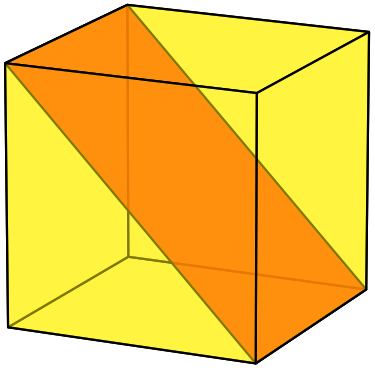
**while** necessary **do**

do one sweep for $U$: act on the first two modes

update $W$: act on the third mode

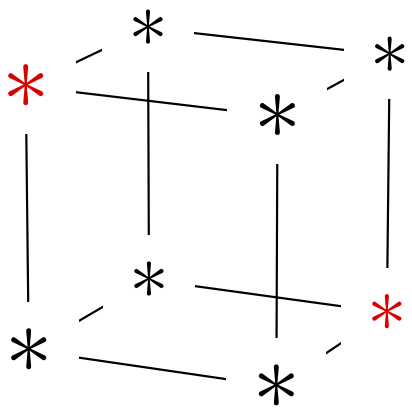
**end while**

# The JADE+ algorithm acts on the third mode

$\Delta \times_3 W =$ 

- take the 'diagonal plane' and normalise its rows
- take an orthogonal approximation of this

# One sweep in the JADE+ algorithm



$$\mathcal{T}_{\text{new}} = \mathcal{T}_{\text{old}} \times_1 Q^{-1} \times_2 Q^{-1}$$

$Q$ is rotation over angle $\theta$

$\tan(\theta)$: root of polynomial of degree 4

One sweep: iterate over $2 \times 2 \times 2$ subtensors

# Experimental comparison

If there is a difference in accuracy,

then it is very small:

as small as the difference in accuracy
due to reducing the number of matrix slices.

Thank you!