Tensor methods

with applications in system identification

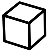
Mariya Ishteva

ADVISE-NUMA

KU LEUVEN

Scalars: $\mathbb{R}$

4 $\quad$ 67

14.4 

3.23 $\quad$ -45.8

$\sqrt{2}$ $\quad$ $-14/8$

$-339/7534$

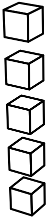
-4.397534

Vectors: $\mathbb{R}^n$

$$\begin{matrix} 4 \\ 15 \\ -45.3 \\ 12.345 \end{matrix}$$

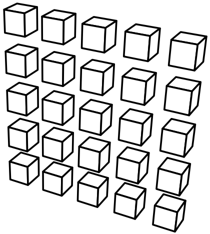
$$\begin{matrix} 65 \\ -15 \\ -23.44 \end{matrix}$$

$$\begin{matrix} 11.1 \\ 12.5 \\ 13.0 \\ 14.4 \\ 12.2 \end{matrix}$$

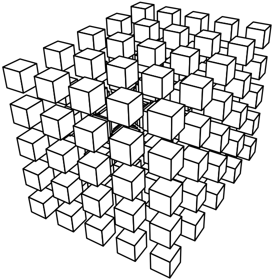


Matrices: $\mathbb{R}^{m \times n}$

10.2	11.1	9.0	9.2	10.5
12.1	12.5	10.2	11.1	12.4
15.1	13.0	10.7	11.7	12.7
19.4	14.4	11.2	12.0	13.1
15.3	12.2	10.9	11.1	12.3

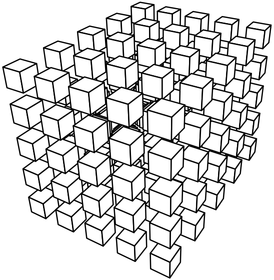


What are tensors?



Elements of the tensor product of N vector spaces

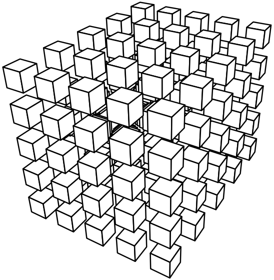
What are tensors?



Elements of the tensor product of N vector spaces

Multi-way arrays

What are tensors?



Elements of the tensor product of N vector spaces

Multi-way arrays

A way to represent data

Tensor methods for system identifiaction

Tensor methods

Block-oriented methods with tensors

Volterra models with tensors

Systems of polynomial equations

Matrix decompositions are being replaced by tensor decompositions

Multi-way (multi-index) data

are naturally processed by tensor methods
instead of (artificially) 'rearranging' them into a matrix.

Multiple data sets

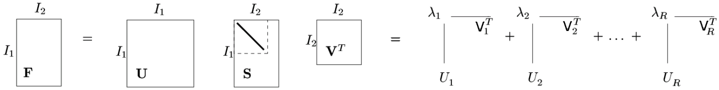
can be processed simultaneously.

"Simple" data

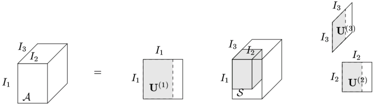
can be tensorized
similarly to Hankelizing vector data.

The ‘tensor SVDs’ generalize the matrix SVD

Singular value decomposition (SVD)



Multilinear SVD



Canonical polyadic decomposition (CPD):

$$\mathcal{A} = [\![\lambda; U^{(1)}, U^{(2)}, U^{(3)}]\!]$$

Tensor decompositions have useful properties

- Interpretability of the factors (uniqueness)
- Suitability for dimensionality reduction
- Ability to solve the 'curse of dimensionality' of high-dimensional data
- Ability to combine multiple data sets (data fusion)
- etc.

Choose the decomposition carefully!
Not all decompositions have all properties.

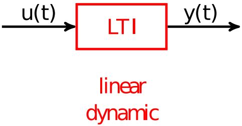
Tensor decompositions are applicable in various domains

- Higher-order statistics
- Chemometrics
- Image processing
- Signal processing
- etc.
- Under-represented in system identification

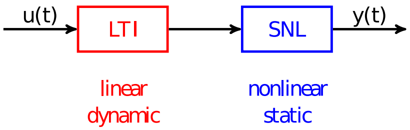
Tensor methods for system identifiaction

- Tensor methods
- Block-oriented methods with tensors
- Volterra models with tensors
- Systems of polynomial equations

Block-oriented models provide a good trade-off between simplicity and descriptive power

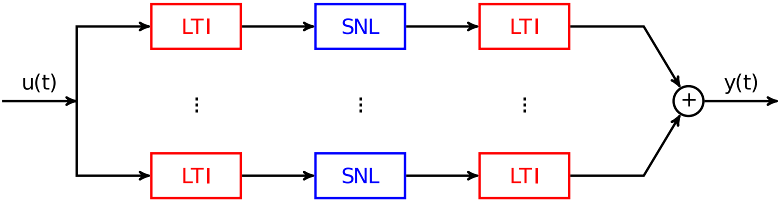


Block-oriented models provide a good trade-off between simplicity and descriptive power

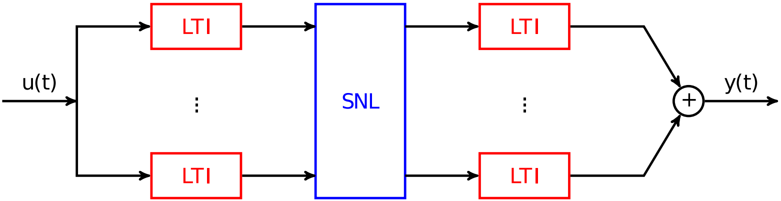


Block-oriented models provide a good trade-off between simplicity and descriptive power

Block-oriented models provide a good trade-off between simplicity and descriptive power

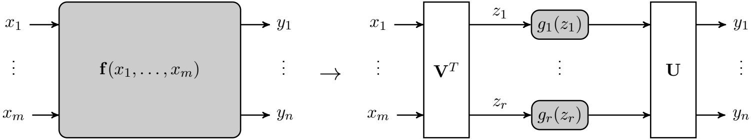


Block-oriented models provide a good trade-off between simplicity and descriptive power



Decoupling multivariate vector functions

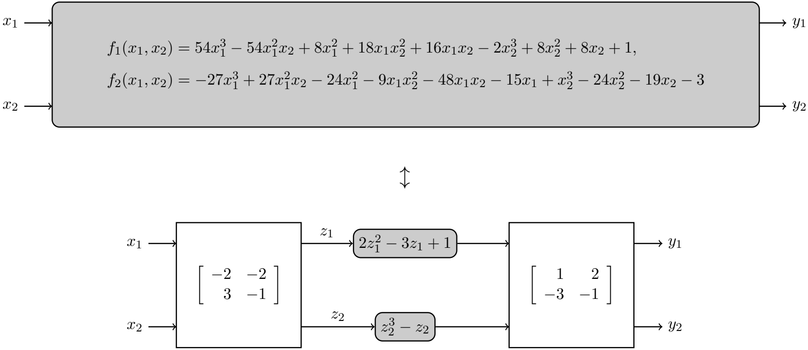
$$\mathbf{f}(\mathbf{x}) = \mathbf{U}\mathbf{g}(\mathbf{V}^T\mathbf{x})$$



Obtain physical insight

Simplification (number of parameters)

A toy example



We compute the NL SVD by a first-order approach

$$\mathbf{f}(\mathbf{x}) = \mathbf{U}\mathbf{g}(\mathbf{V}^T\mathbf{x})$$

We compute the NL SVD by a first-order approach

If $$\mathbf{f}(\mathbf{x}) = \mathbf{U}\mathbf{g}(\mathbf{V}^T\mathbf{x}) \ ,$$

then $$\underbrace{\left[\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right]}_{\mathbf{J_f}(\mathbf{x})} = \mathbf{U} \begin{bmatrix} g_1'(\mathbf{v}_1^T\mathbf{x}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & g_r'(\mathbf{v}_r^T\mathbf{x}) \end{bmatrix} \mathbf{V}^T.$$

We compute the NL SVD by a first-order approach

If $$\mathbf{f}(\mathbf{x}) = \mathbf{U}\mathbf{g}(\mathbf{V}^T\mathbf{x}) \ ,$$

then $$\underbrace{\left[\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right]}_{\mathbf{J}_\mathbf{f}(\mathbf{x})} = \mathbf{U} \begin{bmatrix} g_1'(\mathbf{v}_1^T\mathbf{x}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & g_r'(\mathbf{v}_r^T\mathbf{x}) \end{bmatrix} \mathbf{V}^T.$$

- Collect Jacobian matrices $\mathbf{J}_\mathbf{f}^{(1)}, \mathbf{J}_\mathbf{f}^{(2)}, \mathbf{J}_\mathbf{f}^{(3)}, \mathbf{J}_\mathbf{f}^{(4)}, \mathbf{J}_\mathbf{f}^{(5)}, \ldots$ and diagonalize them simultaneously

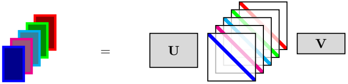
We compute the NL SVD by a first-order approach

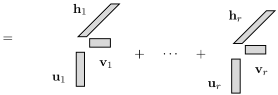
If $$\mathbf{f}(\mathbf{x}) = \mathbf{U}\mathbf{g}(\mathbf{V}^T\mathbf{x})\ ,$$

then $$\underbrace{\left[\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right]}_{\mathbf{J_f}(\mathbf{x})} = \mathbf{U}\begin{bmatrix} g_1'(\mathbf{v}_1^T\mathbf{x}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & g_r'(\mathbf{v}_r^T\mathbf{x}) \end{bmatrix}\mathbf{V}^T.$$

- Collect Jacobian matrices $\mathbf{J}_\mathbf{f}^{(1)}, \mathbf{J}_\mathbf{f}^{(2)}, \mathbf{J}_\mathbf{f}^{(3)}, \mathbf{J}_\mathbf{f}^{(4)}, \mathbf{J}_\mathbf{f}^{(5)}, \ldots$ and diagonalize them simultaneously

  Tool: Canonical Polyadic Decomposition (CPD)



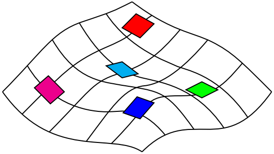


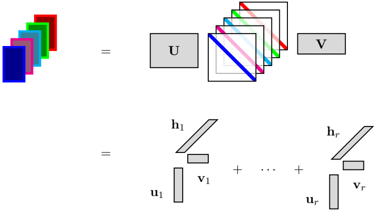
Algorithm

1. Construct tensor of Jacobians
   $\mathcal{J}_{\mathbf{f}} = \left\{ \mathbf{J}_{\mathbf{f}}^{(1)}, \mathbf{J}_{\mathbf{f}}^{(2)}, \mathbf{J}_{\mathbf{f}}^{(3)}, \mathbf{J}_{\mathbf{f}}^{(4)}, \mathbf{J}_{\mathbf{f}}^{(5)}, \ldots \right\}$

2. CPD of $\mathcal{J}_{\mathbf{f}}$ gives $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{H}$

3. Retrieve coefficients of $g_i(\cdot)$ from $\mathbf{y}^{(k)} = \mathbf{U}\left[g_i(\mathbf{v}_i^T \mathbf{x}^{(k)}) \right]$
   (solving linear system)





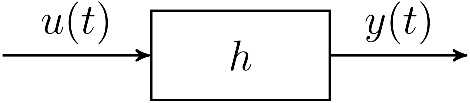
Variations of the main problem are useful in various contexts

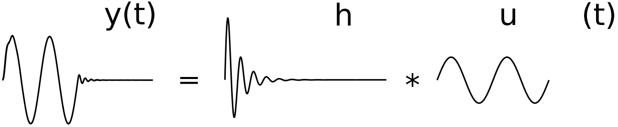
- Scalar functions
  $\rightarrow$ second-order (Hessian) approach
- Uniqueness, noise reduction
  $\rightarrow$ parametrization of the internal functions
  $\rightarrow$ Joint decompositions (combining Jacobians and Hessians)
- Meaningful multivariate internal functions
  $\rightarrow$ block-term decomposition

Tensor methods for system identifiaction

- Tensor methods
- Block-oriented methods with tensors
- Volterra models with tensors
- Systems of polynomial equations

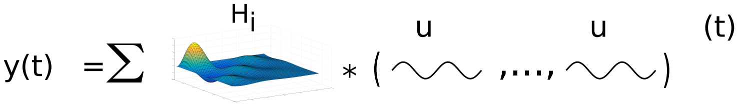
The impulse response completely characterizes *linear* dynamical systems



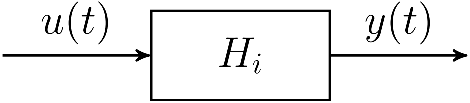


The Volterra kernels completely characterize *nonlinear* dynamical systems

$$u(t) \longrightarrow \boxed{H_i} \longrightarrow y(t)$$



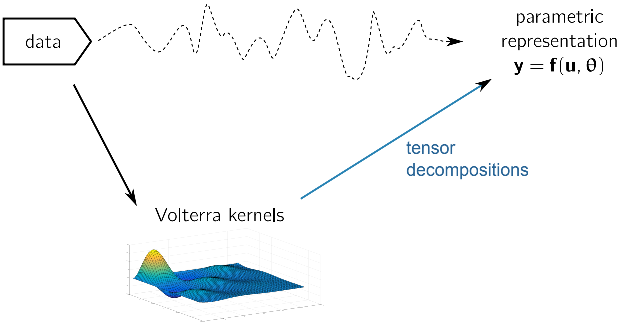
The Volterra kernels completely characterize *nonlinear* dynamical systems



Volterra series are polynomials
of time-shifted input signals

$$y(k) = \sum_{i} \sum_{\tau_1,\ldots,\tau_i} \underbrace{H_i(\tau_1, \ldots, \tau_i)}_{\text{kernels}} \; \underbrace{u(k-\tau_1) \cdots u(k-\tau_i)}_{\text{time-shifted inputs}}$$

To simplify and give meaning to the Volterra kernels we process them by tensor techniques

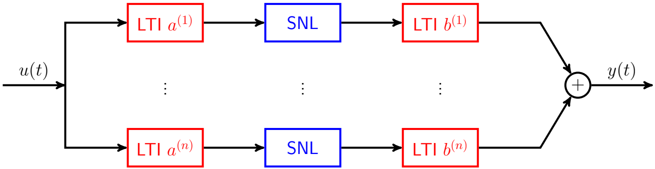


WH systems can be identified through structured tensor decompositions

$$H_3 = [\![\, A\,,\ A\,,\ A\,\mathrm{diag}(b)\,]\!]$$

$$\begin{bmatrix} a_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ & & & a_0 \\ a_m & & & \\ 0 & a_m & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_m \end{bmatrix}$$

Parallel WH systems can be identified through structured tensor decompositions



$$H_3 = [\![A, A, A\,\mathrm{diag}(b)]\!]$$

$$\searrow \quad \left[A^{(1)} \;\middle|\; \cdots \;\middle|\; A^{(n)} \right]$$

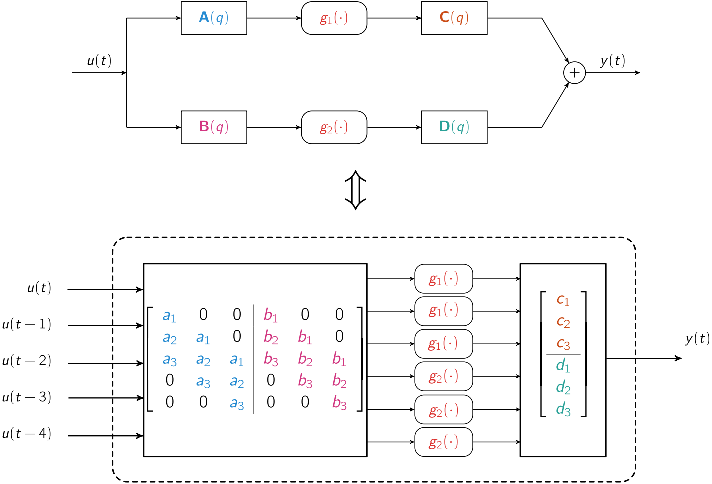
To increase robustness to noise kernels of different orders are decomposed simultaneously

$$
\begin{aligned}
H_3 &= [\![A , A , A , b^T , c_3^T]\!] \\
H_2 &= [\![A , A , b^T , c_2^T]\!] \\
H_1 &= [\![A , b^T , c_1^T]\!]
\end{aligned}
$$

Computation: structured data fusion

Matlab toolbox for tensor computations: Tensorlab

Decoupling Volterra representations



Tensor methods for system identifiaction

Tensor methods

Block-oriented methods with tensors

Volterra models with tensors

Systems of polynomial equations

Can you solve such systems in 5 min?

$$\begin{aligned} x + 3y &= 8 \\ 2x + 6y &= 16 \end{aligned}$$

$$\begin{aligned} x + 3y &= 8 \\ 2x + 6y &= 10 \end{aligned}$$

$$\begin{aligned} x + 3y &= 8 \\ 2x + y &= 6 \\ 3x + 2y &= 10 \end{aligned}$$

$$\begin{aligned} x + 3y + z &= 4 \\ 2x + 6y - z &= 10 \end{aligned}$$

$$\begin{aligned} x + 3y + z + t &= 4 \\ 2x + y + 2z - t &= 3 \\ x - y + 2z - 3t &= 2 \\ -x + 2y - 3z + t &= 1 \end{aligned}$$

$$\begin{aligned} x + 3\sqrt{xy} + \cos(x^2) &= 8 \\ 2x^3 \sin y + y^x &= 6 \end{aligned}$$

Can you solve such systems in 5 min?

$$\begin{aligned} x + 3y &= 8 \\ 2x + 6y &= 16 \end{aligned}$$

$$\begin{aligned} x + 3y &= 8 \\ 2x + 6y &= 10 \end{aligned}$$

$$\begin{aligned} x + 3y &= 8 \\ 2x + y &= 6 \\ 3x + 2y &= 10 \end{aligned}$$

$$\begin{aligned} x + 3y + z &= 4 \\ 2x + 6y - z &= 10 \end{aligned}$$

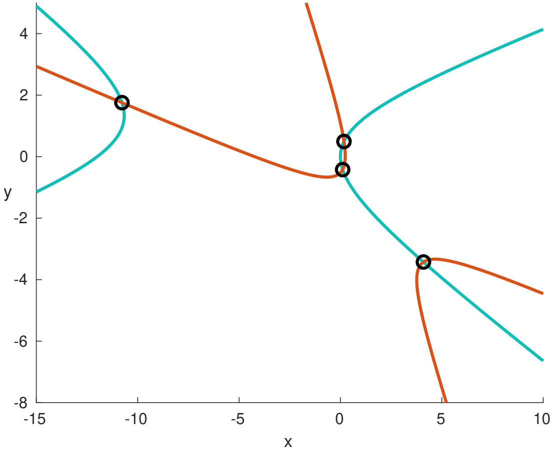
$$\begin{aligned} x + 3y + z + t &= 4 \\ 2x + y + 2z - t &= 3 \\ x - y + 2z - 3t &= 2 \\ -x + 2y - 3z + t &= 1 \end{aligned}$$

$$\begin{aligned} x + 3\sqrt{xy} + \cos\left(x^2\right) &= 8 \\ 2x^3 \sin y + y^x &= 6 \end{aligned}$$

Let us try to solve this system in 5 min

$$
\begin{aligned}
-x^2 + 2xy + 8y^2 - 12x &= 0 \\
2x^2 + 8xy + \tfrac{7}{2}y^2 + 8x - 2y - 2 &= 0
\end{aligned}
$$

Our goal



Step 1: rewrite the system in a tensor form

$$-x^2 + 2xy + 8y^2 - 12x = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -6 \\ 1 & 8 & 0 \\ -6 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$2x^2 + 8xy + \tfrac{7}{2}y^2 + 8x - 2y - 2 = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 4 \\ 4 & \tfrac{7}{2} & -1 \\ 4 & -1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

$$\rightarrow \quad \mathcal{T} \bullet_1 \mathbf{u}^T \bullet_2 \mathbf{u}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{with } \mathbf{u}^T = \begin{bmatrix} x & y & 1 \end{bmatrix}$$

Step 2: rewrite the system by decomposing the tensor

Decompose the tensor $\mathcal{T}$ in (partially-symmetric) rank-1 terms,

$$\mathcal{T} = [\![\mathbf{V}, \mathbf{V}, \mathbf{W}]\!],$$

with $\mathbf{V} \in \mathbb{R}^{3\times r}$ and $\mathbf{W} \in \mathbb{R}^{2\times r}$, ($r$ is the rank of the tensor).

$$\mathbf{V} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & 2 \\ 2 & -2 & 0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} -1 & 1 & 2 \\ \frac{1}{2} & -1 & 1 \end{bmatrix}.$$

$$\mathcal{T} \bullet_1 \mathbf{u}^T \bullet_2 \mathbf{u}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \rightarrow \quad [\![\mathbf{u}^T\mathbf{V}, \mathbf{u}^T\mathbf{V}, \mathbf{W}]\!] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Next: two subproblems (two systems of linear equations!)

$$\mathcal{T} \bullet_1 \mathbf{u}^T \bullet_2 \mathbf{u}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \rightarrow \quad [\![\underbrace{\mathbf{u}^T \mathbf{V}}_{\mathbf{z}^T}, \mathbf{u}^T \mathbf{V}, \mathbf{W}]\!] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- Given $\mathbf{W}$, find $\mathbf{z}$ (Step 3)
- Given $\mathbf{V}$, find $\mathbf{u}$ (Step 4)

The rank of $\mathcal{T}$ is 3.

(Typical ranks of a $3 \times 3 \times 2$ tensor: $\{3, 4\}$ in $\mathbb{R}$ and 3 in $\mathbb{C}$.)

Step 3: find $\mathbf{z}$

$$[\![\underbrace{\mathbf{u}^T\mathbf{V}}_{\mathbf{z}^T}, \mathbf{u}^T\mathbf{V}, \mathbf{W}]\!] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- $\mathbf{W}\mathbf{z}.^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ($\mathbf{z}.^2$ contains the squared elements of $\mathbf{z}$)
- Linear system of equations for $\mathbf{z}.^2$

$$\mathbf{z}.^2 = \begin{bmatrix} 0.8242 \\ 0.5494 \\ 0.1374 \end{bmatrix}$$

- Four essentially different solutions for $\mathbf{z}$

$$\mathbf{z}^{(1)} = \begin{bmatrix} 0.9078 \\ 0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(2)} = \begin{bmatrix} -0.9078 \\ 0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(3)} = \begin{bmatrix} 0.9078 \\ -0.7412 \\ 0.3706 \end{bmatrix}, \mathbf{z}^{(4)} = \begin{bmatrix} 0.9078 \\ 0.7412 \\ -0.3706 \end{bmatrix}$$

Step 4: find $\mathbf{u}$ (find x and y)

$$\mathbf{z}^T = \mathbf{u}^T\mathbf{V} \quad \rightarrow \quad \mathbf{V}^T\mathbf{u} = \mathbf{z}$$

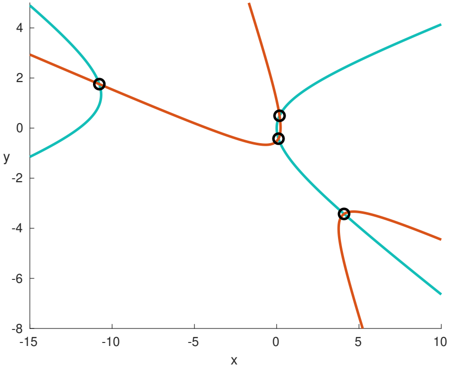
$$\mathbf{u}^{(1)} = \begin{bmatrix} 0.5497 \\ -0.0895 \\ -0.0510 \end{bmatrix} = -0.0510 \begin{bmatrix} -10.7766 \\ 1.7553 \\ 1 \end{bmatrix} \quad \rightarrow \begin{bmatrix} x^{(1)} \\ y^{(1)} \end{bmatrix} = \begin{bmatrix} -10.7766 \\ 1.7553 \end{bmatrix}$$

$$\mathbf{u}^{(2)} = \begin{bmatrix} -0.0555 \\ 0.2131 \\ -0.5049 \end{bmatrix} = -0.5049 \begin{bmatrix} 0.1100 \\ -0.4220 \\ 1 \end{bmatrix} \quad \rightarrow \begin{bmatrix} x^{(2)} \\ y^{(2)} \end{bmatrix} = \begin{bmatrix} 0.1100 \\ -0.4220 \end{bmatrix}$$

$$\mathbf{u}^{(3)} = \begin{bmatrix} 0.0555 \\ 0.1575 \\ 0.3196 \end{bmatrix} = 0.3196 \begin{bmatrix} 0.1737 \\ 0.4929 \\ 1 \end{bmatrix} \quad \rightarrow \begin{bmatrix} x^{(3)} \\ y^{(3)} \end{bmatrix} = \begin{bmatrix} 0.1737 \\ 0.4929 \end{bmatrix}$$

$$\mathbf{u}^{(4)} = \begin{bmatrix} 0.5497 \\ -0.4602 \\ 0.1343 \end{bmatrix} = 0.1343 \begin{bmatrix} 4.0929 \\ -3.4263 \\ 1 \end{bmatrix} \quad \rightarrow \begin{bmatrix} x^{(4)} \\ y^{(4)} \end{bmatrix} = \begin{bmatrix} 4.0929 \\ -3.4263 \end{bmatrix}$$

The solutions



$$\begin{bmatrix} x^{(1)} \\ y^{(1)} \end{bmatrix} = \begin{bmatrix} -10.7766 \\ 1.7553 \end{bmatrix}$$

$$\begin{bmatrix} x^{(2)} \\ y^{(2)} \end{bmatrix} = \begin{bmatrix} 0.1100 \\ -0.4220 \end{bmatrix}$$

$$\begin{bmatrix} x^{(3)} \\ y^{(3)} \end{bmatrix} = \begin{bmatrix} 0.1737 \\ 0.4929 \end{bmatrix}$$

$$\begin{bmatrix} x^{(4)} \\ y^{(4)} \end{bmatrix} = \begin{bmatrix} 4.0929 \\ -3.4263 \end{bmatrix}$$

Solving a system of polynomial equations via tensor decomposition

Given: 2 polynomial equations of degree 2 (in 2 variables)
Find: The solutions $(x^{(i)}, y^{(i)}), i = 1, ..., 4$ of the system

1. Reformulate the problem as $\mathcal{T} \bullet_1 \mathbf{u}^T \bullet_2 \mathbf{u}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
2. Decompose the tensor $\mathcal{T}$ in rank-1 terms: $\mathcal{T} = [\![\mathbf{V}, \mathbf{V}, \mathbf{W}]\!]$.
3. Solve the linear system $\mathbf{W}(\mathbf{z}.^2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ for $\mathbf{z}.^2$.

   Find the 4 (essentially different) solutions $\mathbf{z}^{(i)}$, $i = 1, ..., 4$.
4. Solve the linear systems $\mathbf{V}^T \mathbf{u}^{(i)} = \mathbf{z}^{(i)}$ for $\mathbf{u}^{(i)}$.
   Rescale $\mathbf{u}^{(i)}$ (the last entries become 1s) and remove the 1s.
   $\rightarrow (x^{(i)}, y^{(i)})$, $i = 1, ..., 4$.

Generalizations

1. Roots: $\mathbb{R}$, $\mathbb{C}$, roots at infinity
2. Polynomials of higher degree
3. More unknowns and more equations

Generalizations

1. Roots: $\mathbb{R}$, $\mathbb{C}$, roots at infinity: OK

2. Polynomials of higher degree

3. More unknowns and more equations

Generalizations

1. Roots: $\mathbb{R}$, $\mathbb{C}$, roots at infinity: OK
2. Polynomials of higher degree: 4th order tensor
3. More unknowns and more equations

Generalizations

1. Roots: $\mathbb{R}$, $\mathbb{C}$, roots at infinity: OK
2. Polynomials of higher degree: 4th order tensor
3. More unknowns and more equations: longer $\mathbf{u}$

Generalizations

1. Roots: $\mathbb{R}$, $\mathbb{C}$, roots at infinity: OK
2. Polynomials of higher degree: 4th order tensor
3. More unknowns and more equations: longer $\mathbf{u}$

   However, the rank of the tensor might increase! (2. and 3.)

Systems with tensors of higher rank: future work

Problem if r is large: since $\mathbf{W} \in \mathbb{R}^{n \times r}$,
the solution of $\mathbf{W}\mathbf{z}.^{d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ lies in a higher dimensional space.

A possible direction to consider:
reformulate the problem as

$$\mathbf{W}(\mathbf{V} \odot \mathbf{V})^{T}(\mathbf{u} \otimes \mathbf{u}) = \mathbf{0}$$

and solve for $\mathbf{u} \otimes \mathbf{u} \ldots$

Homework: solve in 5 min!

$$\begin{aligned}
x^2 + 4xy + y^2 - 2x + 2y &= 0 \\
x^2 + 5xy + 6x - 2y + 1 &= 0
\end{aligned}$$

Tensor methods for system identifiaction

Tensor methods

Block-oriented methods with tensors

Volterra models with tensors

Systems of polynomial equations

Joint work with

Philippe Dreesen

Johan Schoukens

Konstantin Usevich

David Westwick

Gabriel Hollander, Jeroen De Geeter, Thomas Goossens

Tensor methods are useful

For multi-way (multi-index) data

Use tensor methods instead of (artificially) ‘rearranging’ the data into a matrix.

For multiple data sets

Process related data sets simultaneously.

For other data

Tensorize ‘simple’ data in the same way as, for example, vector data can be Hankelized.

Tensor methods

with applications in system identification

Mariya Ishteva

ADVISE-NUMA