Decompositions of Higher-Order Tensors: Concepts and Computation

Lieven De Lathauwer

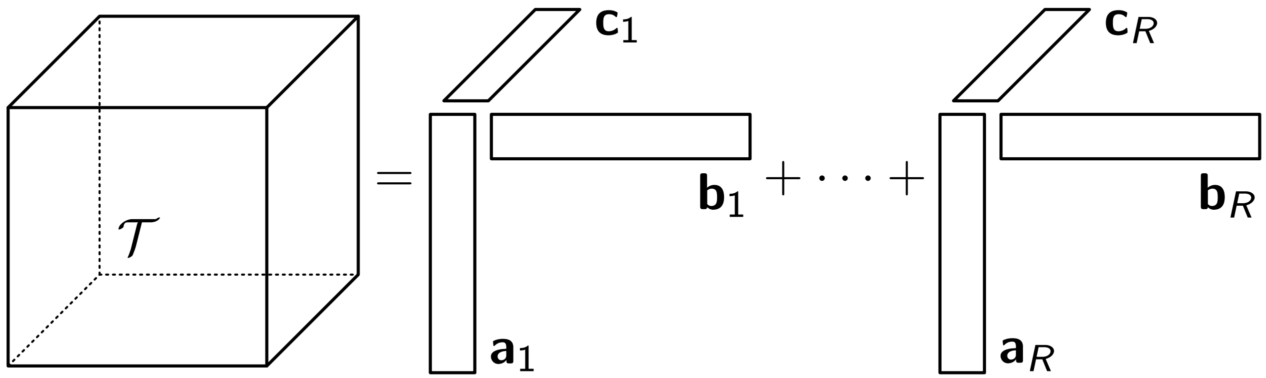
KU Leuven
Belgium

Lieven.DeLathauwer@kuleuven-kulak.be

Canonical Polyadic Decomposition

Rank: minimal number of rank-1 terms [*Hitchcock, 1927*]

Canonical Polyadic Decomposition (CPD): decomposition in minimal number of rank-1 terms [*Harshman '70*], [*Carroll and Chang '70*]



- Unique under mild conditions on number of terms and differences between terms
- Orthogonality (triangularity, . . .) not required (but may be imposed)

Overview

- Basics: Rank and Canonical Polyadic Decomposition
- Conceptual advances: CPD uniqueness
- Conceptual advances: more general decompositions and variants
- Computational advances: numerical optimization

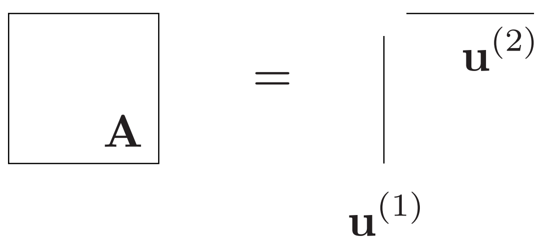
Rank-1 tensor

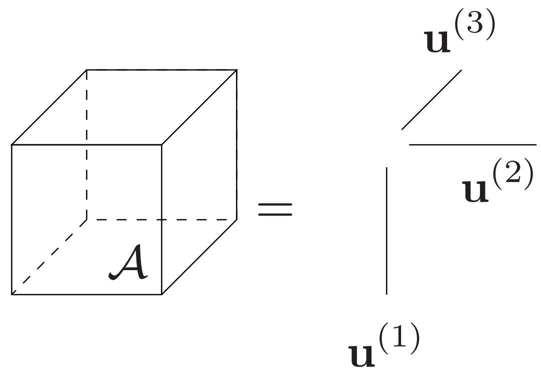
- Rank-1 matrix: tensor (outer) product of 2 vectors $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$:

$$\begin{aligned} a_{i_1 i_2} &= u^{(1)}_{i_1} u^{(2)}_{i_2} \\ \mathbf{A} &= \mathbf{u}^{(1)} \cdot {\mathbf{u}^{(2)}}^T \equiv \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \end{aligned}$$

- Rank-1 tensor: tensor (outer) product of N vectors $\mathbf{u}^{(1)}$, $\mathbf{u}^{(2)}$, . . . , $\mathbf{u}^{(N)}$:

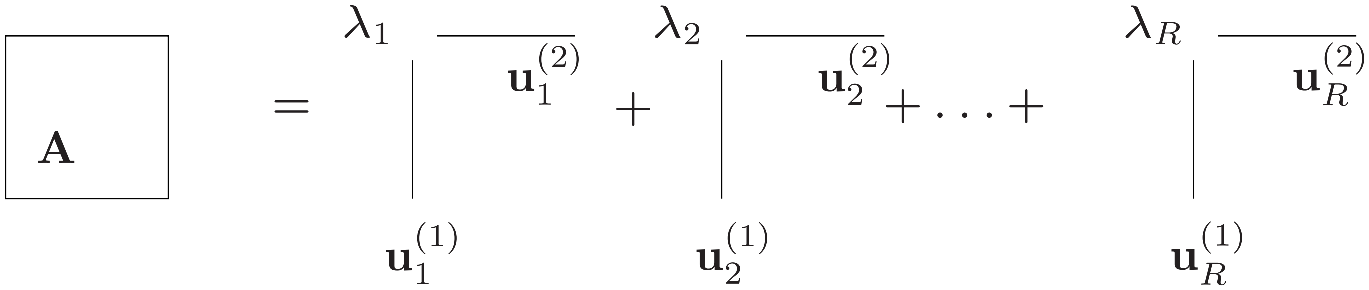
$$\begin{aligned} a_{i_1 i_2 \ldots i_N} &= u^{(1)}_{i_1} u^{(2)}_{i_2} \ldots u^{(N)}_{i_N} \\ \mathcal{A} &= \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ \ldots \circ \mathbf{u}^{(N)} \end{aligned}$$



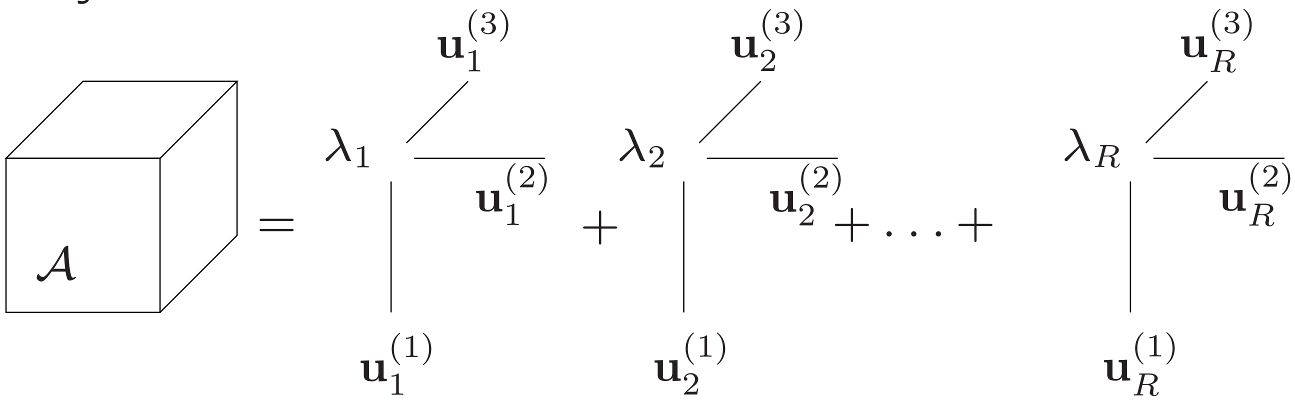


Rank of a tensor

- The rank R of a matrix $\mathbf{A}$ is minimal number of rank-1 matrices that yield $\mathbf{A}$ in a linear combination.



- The rank R of an Nth-order tensor $\mathcal{A}$ is the minimal number of rank-1 tensors that yield $\mathcal{A}$ in a linear combination.



[*Hitchcock, 1927*]

Rank and dimension

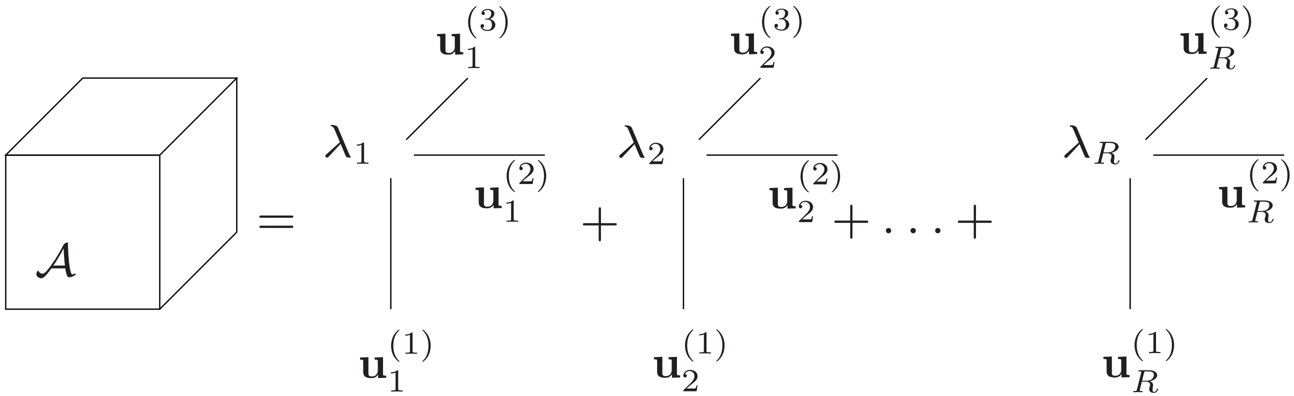
Matrices:

The rank of a $(K \times K)$ matrix is at most equal to K

Tensors:

The rank of a $(K \times K \times \ldots \times K)$ tensor can be greater than K

Partial explanation: number of free tensor parameters: K^N
number of parameters in expansion: NKR



Rank and multilinear rank: $R \geqslant \max(R_1, R_2, \ldots, R_N)$

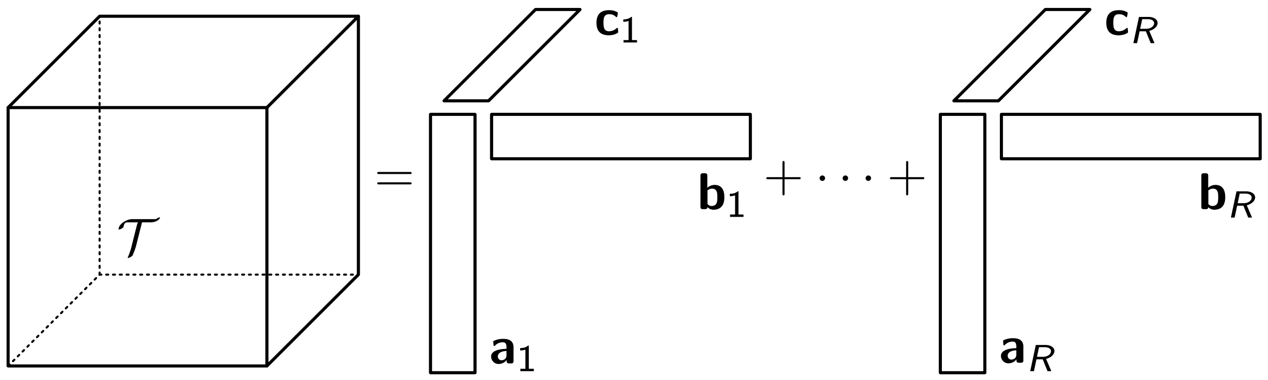
Overview

- Basics: Rank and Canonical Polyadic Decomposition
- Conceptual advances: CPD uniqueness
- Conceptual advances: more general decompositions and variants
- Computational advances: numerical optimization

Canonical Polyadic Decomposition

Rank: minimal number of rank-1 terms [*Hitchcock, 1927*]

Canonical Polyadic Decomposition (CPD): decomposition in minimal number of rank-1 terms [*Harshman '70*], [*Carroll and Chang '70*]

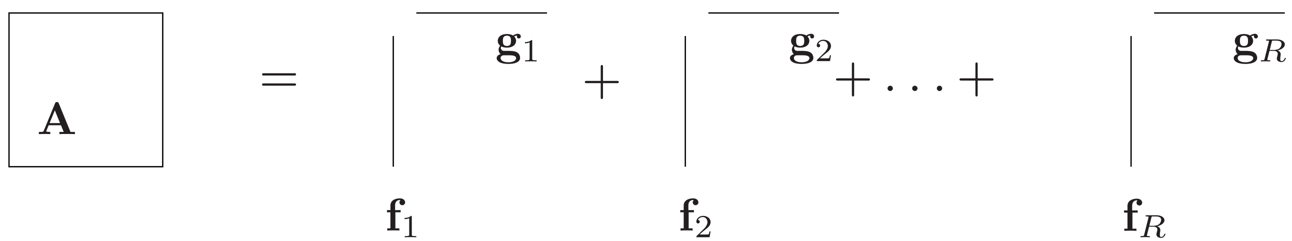


- Unique under mild conditions on number of terms and differences between terms
- Orthogonality (triangularity, . . .) not required (but may be imposed)

Factor Analysis and Blind Source Separation

- Decompose a data matrix in rank-1 terms that can be interpreted
  E.g. statistics, telecommunication, biomedical applications, chemometrics, data analysis, . . .

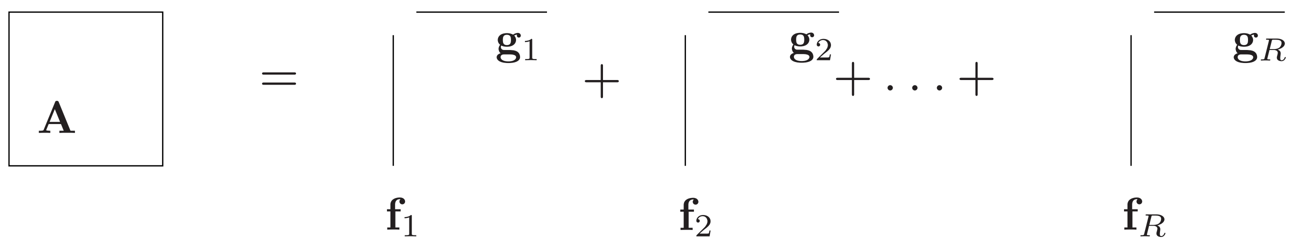
$$\mathbf{A} = \mathbf{F} \cdot \mathbf{G}^T$$



- $\mathbf{F}$: mixing matrix
  $\mathbf{G}$: source signals

- Decompose a data matrix in rank-1 terms that can be interpreted

$$\mathbf{A} = \mathbf{F} \cdot \mathbf{G}^T$$



- Problem: decomposition in rank-1 terms is not unique

$$\begin{aligned} \mathbf{A} &= (\mathbf{F}\mathbf{M}) \cdot (\mathbf{M}^{-1}\mathbf{G}^T) \\ &= \tilde{\mathbf{F}} \cdot \tilde{\mathbf{G}}^T \end{aligned}$$

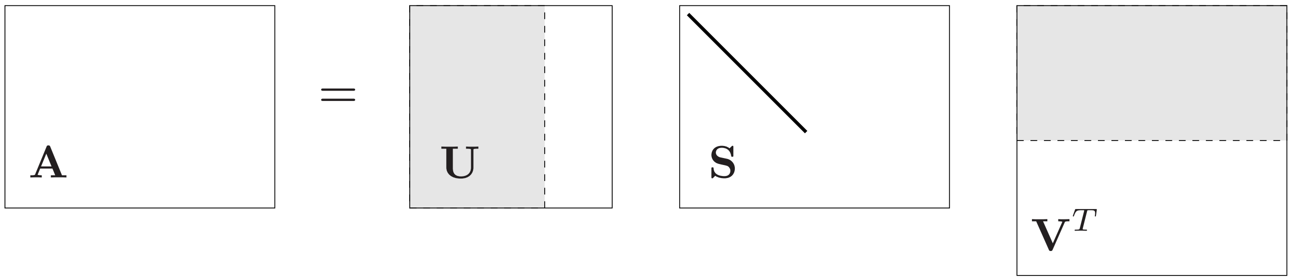
What about SVD?

- SVD is unique
- . . . thanks to orthogonality constraints

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \sum_{r=1}^{R} s_{rr} \mathbf{u}_r \mathbf{v}_r^T$$

$\mathbf{U}$, $\mathbf{V}$ orthogonal, $\mathbf{S}$ diagonal

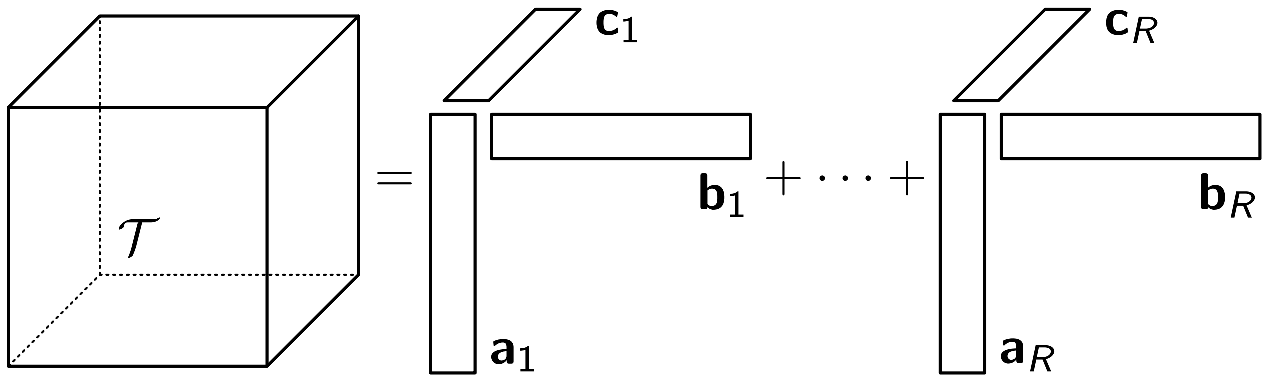
- Whether these constraints make sense, depends on the application
- SVD is great for dimensionality reduction
  best rank-R approximation $\leftarrow$ truncated SVD



Canonical Polyadic Decomposition

Rank: minimal number of rank-1 terms [*Hitchcock, 1927*]

Canonical Polyadic Decomposition (CPD): decomposition in minimal number of rank-1 terms [*Harshman '70*], [*Carroll and Chang '70*]



- Unique under mild conditions on number of terms and differences between terms
- Orthogonality (triangularity, . . .) not required (but may be imposed)

Uniqueness: Kruskal's Theorem

Rank: at least one set of $r_{\mathbf{A}}$ columns is independent

K-rank: every set of $k_{\mathbf{A}}$ columns is independent ($k_{\mathbf{A}} \leqslant r_{\mathbf{A}}$) ($k_{\mathbf{A}} + 1$ is spark)

Theorem:

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2$$

$\rightarrow r_{\mathcal{T}} = R$ and CPD is unique

[*Kruskal '77*]

Generic: $\mathbf{A}(I \times R) \quad \mathbf{B}(J \times R) \quad \mathbf{C}(K \times R)$
CPD is unique for R bounded by I, J, K as in

$$\min(I, R) + \min(J, R) + \min(K, R) \geqslant 2R + 2$$

New conditions

Kruskal-type corollary:

Let at least two of the following conditions hold:

$$\begin{cases} k_{\mathbf{A}} + r_{\mathbf{B}} + r_{\mathbf{C}} \geq & 2R+2 \\ r_{\mathbf{A}} + k_{\mathbf{B}} + r_{\mathbf{C}} \geq & 2R+2 \\ r_{\mathbf{A}} + r_{\mathbf{B}} + k_{\mathbf{C}} \geq & 2R+2 \end{cases}$$

$\rightarrow r_{\mathcal{T}} = R$ and CPD is unique

[*Domanov, DL '12*]

Uniqueness: C has full column rank

CPD: $\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \in \mathbb{C}^{I \times J \times K}$ $\qquad \mathbf{T}_{[1,2;3]} = (\mathbf{A} \odot \mathbf{B}) \cdot \mathbf{C}^T \in \mathbb{C}^{IJ \times K}$

e.g. $\mathbf{C}$-mode is sample mode

Khatri-Rao product second compound matrices:

$$\mathbf{U} = C_2(\mathbf{A}) \odot C_2(\mathbf{B}) \in \mathbb{C}^{\frac{I(I-1)}{2}\frac{J(J-1)}{2} \times \frac{R(R-1)}{2}}$$

$$u_{i_1 i_2 j_1 j_2 r_1 r_2} = \begin{vmatrix} a_{i_1 r_1} & a_{i_2 r_1} \\ a_{i_1 r_2} & a_{i_2 r_2} \end{vmatrix} \cdot \begin{vmatrix} b_{j_1 r_1} & b_{j_2 r_1} \\ b_{j_1 r_2} & b_{j_2 r_2} \end{vmatrix}$$

$$1 \leqslant i_1 < i_2 \leqslant I \quad 1 \leqslant j_1 < j_2 \leqslant J \quad 1 \leqslant r_1 < r_2 \leqslant R$$

Theorem: if $\mathbf{U}$ and $\mathbf{C}$ have full column rank, then CPD is unique

(proof is constructive)

[*Jiang and Sidiropoulos, '04*], [*DL '06*]

Uniqueness: $\mathbf{C}$ has full column rank (2)

Theorem: if $\mathbf{U} \in \mathbb{C}^{\frac{I(I-1)}{2}\frac{J(J-1)}{2} \times \frac{R(R-1)}{2}}$ and $\mathbf{C} \in \mathbb{C}^{K \times R}$ have full column rank, then CPD is unique

Generic: CPD is unique for R bounded by I, J, K as in

$$\frac{I(I-1)}{2}\frac{J(J-1)}{2} \geqslant \frac{R(R-1)}{2} \quad \text{and} \quad K \geqslant R$$

Approximately: $\frac{IJ}{\sqrt{2}} \geqslant R \quad K \geqslant R$

Compare to Kruskal:

$$\min(I,R) + \min(J,R) \geqslant R + 2 \quad \text{and} \quad K \geqslant R$$

Recent results

Unifying theory

Constructive proof

Algorithm for Kruskal's condition (and beyond)

[*Domanov, DL, '12*], [*Domanov, DL, '13*]

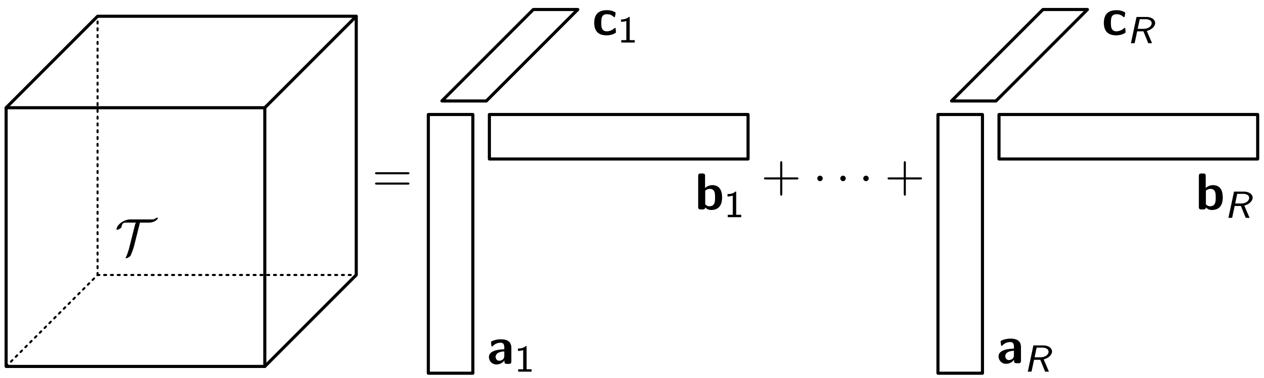
Overview

- Basics: Rank and Canonical Polyadic Decomposition
- Conceptual advances: CPD uniqueness
- Conceptual advances:
  - Block terms
  - Coupled decompositions
  - Constraints
- Computational advances: numerical optimization

Canonical Polyadic Decomposition

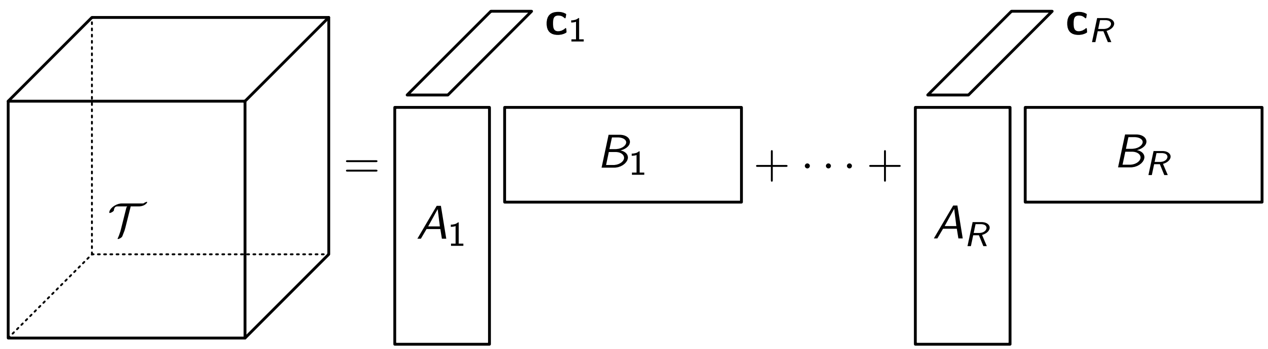
Rank: minimal number of rank-1 terms [*Hitchcock, 1927*]

Canonical Polyadic Decomposition (CPD): decomposition in minimal number of rank-1 terms [*Harshman '70*], [*Carroll and Chang '70*]



- Unique under mild conditions on number of terms and differences between terms
- Orthogonality (triangularity, . . .) not required (but may be imposed)

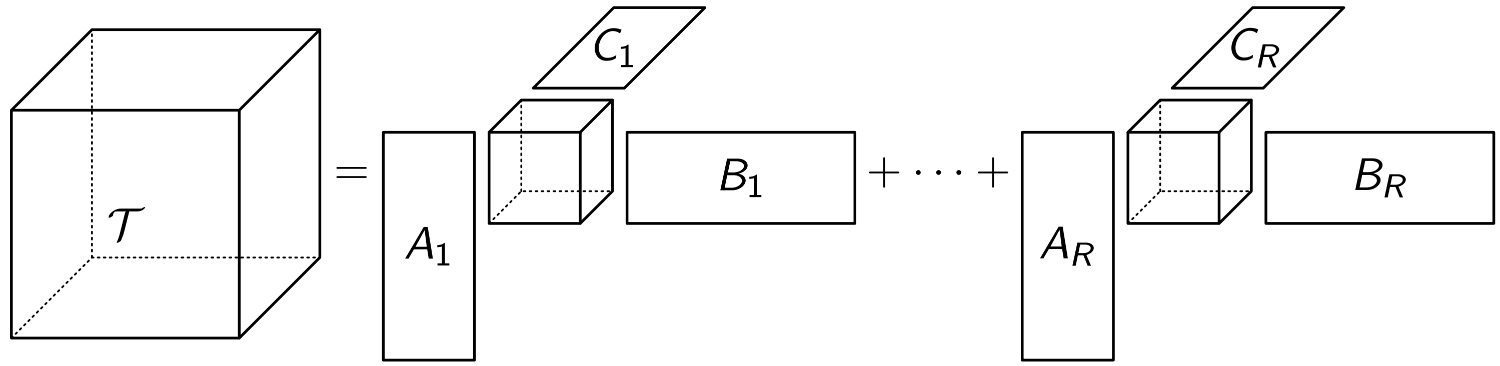
Decomposition in rank-$(L, L, 1)$ terms



Unique under mild conditions

[*DL '08*]

Decomposition in rank-(R_1, R_2, R_3) terms



Unique under mild conditions

Rank-1 term $\sim$ data atom
Block term $\sim$ data molecule

[*DL '08*]

Constraints

Examples: orthogonality [*Sørensen and DL '12*]
nonnegativity [*Cichocki et al. '09*]
Vandermonde [*Sørensen and DL '12*]
independence [*De Vos et al. '12*]
...

Not needed for uniqueness in tensor case

Pro: relaxed uniqueness conditions
easier interpretation
no degeneracy (NN, orthogonality)
higher accuracy

Depending on type of constraints, lower or higher computational cost

Coupled matrix/tensor decompositions

One or more matrices

One or more tensors

Symmetric and nonsymmetric

One or more factors shared (or parts of factors, or generators)

Constraints (orthogonal, nonnegative, exponential, constant modulus, polynomial, rational, Toeplitz, Hankel, . . .)

Data fusion

Overview

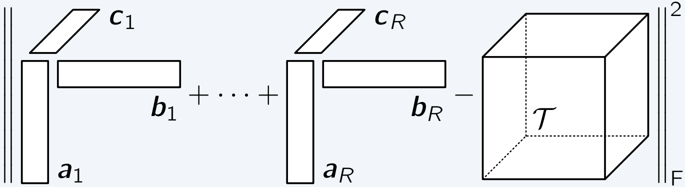
- Basics: Rank and Canonical Polyadic Decomposition
- Conceptual advances: CPD uniqueness
- Conceptual advances: more general decompositions and variants
- Computational advances:
  - Optimization of complex variables
  - Numerical optimization
  - Exact line and plane search
  - Framework for (constrained) coupled decompositions

KU LEUVEN

Between linear and nonlinear: numerical computation of tensor decompositions

Laurent Sorber, Marc Van Barel and Lieven De Lathauwer

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}$$



where $z^\top := \begin{bmatrix} a_1^\top & \cdots & a_R^\top & b_1^\top & \cdots & b_R^\top & c_1^\top & \cdots & c_R^\top \end{bmatrix}$

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x})$$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

- f is not differentiable w.r.t. z
  No real-valued functions are analytic in complex z!

$$\underset{\mathbf{z} \in \mathbb{C}^n}{\text{minimize}} \quad f(\mathbf{z}, \overline{\mathbf{z}})$$

- f is not differentiable w.r.t. $\mathbf{z}$
  No real-valued functions are analytic in complex $\mathbf{z}$!
- Defacto solution is to minimize $f(\mathbf{z}_R)$ where $\mathbf{z}_R := \begin{bmatrix} \mathrm{Re}\{\mathbf{z}\} \\ \mathrm{Im}\{\mathbf{z}\} \end{bmatrix}$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

- f is not differentiable w.r.t. z
  No real-valued functions are analytic in complex z!
- Defacto solution is to minimize $f(z_R)$ where $z_R := \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$
- Alternatively, use complex optimization [S,VB,DL]

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

and define the complex gradient as

$$\frac{\partial f}{\partial z_C} := J^{-\mathsf{T}} \cdot \frac{\partial f}{\partial z_R} = \frac{1}{2} \begin{bmatrix} \frac{\partial f}{\partial \mathrm{Re}\{z\}} - \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \\ \frac{\partial f}{\partial \mathrm{Re}\{z\}} + \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \end{bmatrix} =: \begin{bmatrix} \frac{\partial f}{\partial z} \\ \frac{\partial f}{\partial \overline{z}} \end{bmatrix}$$

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

and define the complex gradient as

$$\frac{\partial f}{\partial z_C} := J^{-\mathsf{T}} \cdot \frac{\partial f}{\partial z_R} = \frac{1}{2} \begin{bmatrix} \frac{\partial f}{\partial \mathrm{Re}\{z\}} - \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \\ \frac{\partial f}{\partial \mathrm{Re}\{z\}} + \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \end{bmatrix} =: \begin{bmatrix} \frac{\partial f}{\partial z} \\ \frac{\partial f}{\partial \overline{z}} \end{bmatrix}$$

Real Taylor series

$$f(\boldsymbol{z}^{(k)}) + \boldsymbol{p}_R^\mathsf{T} \qquad \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^\mathsf{T} \qquad \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^\mathsf{T}} \qquad \boldsymbol{p}_R$$

Real Taylor series

$$f(\boldsymbol{z}^{(k)}) + \boldsymbol{p}_R^\mathsf{T} \cdot J^\mathsf{T} J^{-\mathsf{T}} \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^\mathsf{T} \cdot J^\mathsf{T} J^{-\mathsf{T}} \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^\mathsf{T}} \cdot J^\mathsf{T} J^{-\mathsf{T}} \cdot \boldsymbol{p}_R$$

Real Taylor series

$$f(z^{(k)}) + \boldsymbol{p}_R^\top \cdot J^\top J^{-\top} \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^\top \cdot J^\top J^{-\top} \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^\top} \cdot J^\top J^{-\top} \cdot \boldsymbol{p}_R$$

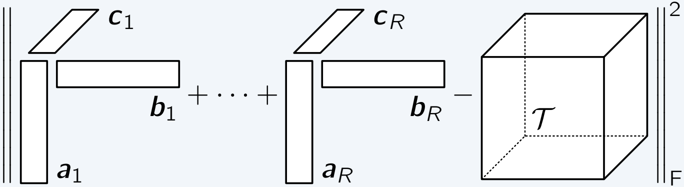
Complex Taylor series

$$f(z^{(k)}) + \boldsymbol{p}_C^\top \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_C} + \boldsymbol{p}_C^\top \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_C \partial \boldsymbol{z}_C^\top} \cdot \boldsymbol{p}_C$$

Complex Optimization Toolbox (COT) for MATLAB
esat.kuleuven.be/sista/cot

- Generalized nonlinear optimization
  `minf_lbfgs`, `minf_lbfgsdl`, `minf_ncg`
- Generalized nonlinear least squares
  `nls_gndl`, `nls_lm`, `nls_gncgs`, `nlsb_gndl`
- Complex differentiation and Moré–Thuente line search
  `deriv`, `ls_mt`

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}$$



where $\boldsymbol{z}^\top := \begin{bmatrix} \boldsymbol{a}_1^\top & \cdots & \boldsymbol{a}_R^\top & \boldsymbol{b}_1^\top & \cdots & \boldsymbol{b}_R^\top & \boldsymbol{c}_1^\top & \cdots & \boldsymbol{c}_R^\top \end{bmatrix}$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(z) - \mathcal{T}\|_{\mathsf{F}}^2$$

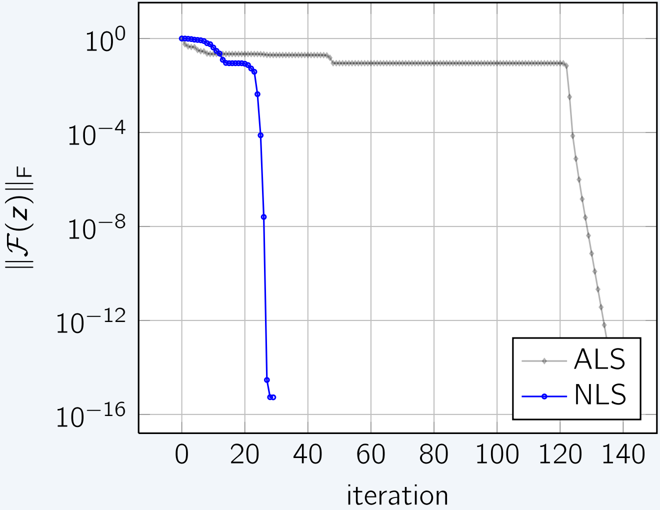
where $\mathcal{M}$ is multilinear

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{F}(z)\|_{\mathsf{F}}^2$$

where $\mathcal{F}$ is multilinear

- canonical polyadic decomposition (CPD),
- low multilinear rank approximation (LMLRA),
- block term decompositions (BTD),
- support tensor machines (STM),
- coupled tensor-matrix factorizations (CTMF),
- ...

CPD of a $9 \times 9 \times 9 \times 9 \times 9$ tensor of rank 11



The step is computed as

$$\boldsymbol{p}^* = -H^{-1}\boldsymbol{g}$$

$f(z, \overline{z}) := \frac{1}{2}\|\mathcal{F}(z)\|_{\mathrm{F}}^2$ is the objective function
$\boldsymbol{g} := 2\frac{\partial f}{\partial \overline{z}}$ is the scaled conjugate cogradient
H := is (an approximation of) the complex Hessian

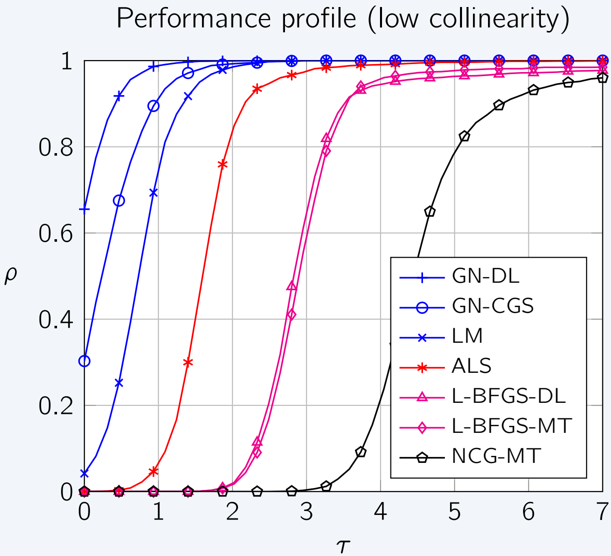
Where H is

- a diagonal plus low-rank matrix in quasi-Newton
- $J^{\mathrm{H}}J$ in NLS and $J := \frac{\partial \mathcal{F}}{\partial \boldsymbol{z}^{\mathrm{T}}}$

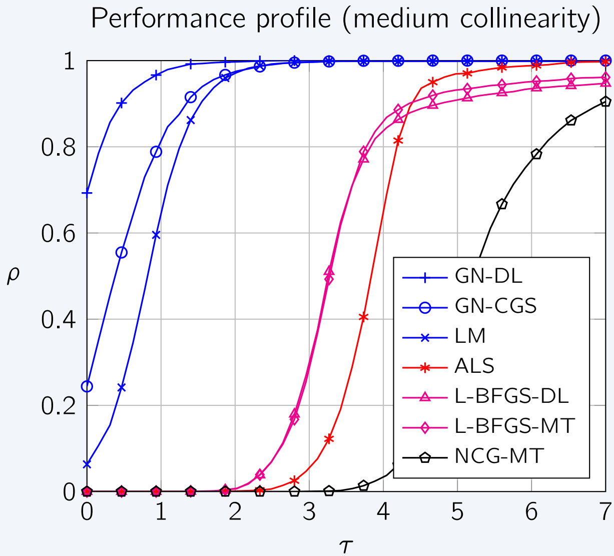
- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - N^2R^2 times more flop/iteration than ALS

- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - $N^2 R^2$ times more flop/iteration than ALS
- Exploit rank-one and diagonal block structure in $J^{\mathrm{H}} J$ to obtain a fast inexact NLS algorithm [S,VB,DL]
  - Same memory cost as ALS
  - Same flop/iteration as ALS for large tensors

- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - N^2R^2 times more flop/iteration than ALS
- Exploit rank-one and diagonal block structure in $J^{\mathrm{H}}J$ to obtain a fast inexact NLS algorithm [S,VB,DL]
  - Same memory cost as ALS
  - Same flop/iteration as ALS for large tensors
- Additional benefits (compared to ALS)
  - Almost “embarrassingly” parallel
    Can theoretically achieve peak performance on GPUs
  - Robust performance on difficult decompositions



Performance profile (low collinearity)
1
0.8
0.6
0.4
0.2
0
ρ
0
1
2
3
4
5
6
7
τ
GN-DL
GN-CGS
LM
ALS
L-BFGS-DL
L-BFGS-MT
NCG-MT


Performance profile (medium collinearity)
ρ
τ
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4
5
6
7
GN-DL
GN-CGS
LM
ALS
L-BFGS-DL
L-BFGS-MT
NCG-MT

Tensorlab — a MATLAB toolbox for tensor decompositions
`esat.kuleuven.be/sista/tensorlab`

- Elementary operations on tensors
  Multicore-aware and profiler tuned
- Tensor decompositions with structure and/or symmetry
  CPD, LMLRA, MLSVD, block term decompositions
- Global minimization of bivariate polynomials
  Exact line and plane search for tensor optimization
- Cumulants, tensor visualization, estimating a tensor's rank or multilinear rank, ...

Tensorlab — a MATLAB toolbox for tensor decompositions
`esat.kuleuven.be/sista/tensorlab`

- Elementary operations on tensors
  Multicore-aware and profiler tuned
- Tensor decompositions with structure and/or symmetry
  CPD, LMLRA, MLSVD, block term decompositions
- Global minimization of bivariate polynomials
  Exact line and plane search for tensor optimization
- Cumulants, tensor visualization, estimating a tensor's rank or multilinear rank, . . .

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(z + \alpha \Delta z) - \mathcal{T}\|_{\mathsf{F}}^2 \tag{LS}$$

$$\underset{\alpha,\gamma}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\gamma z + \alpha \Delta z) - \mathcal{T}\|_{\mathsf{F}}^2 \tag{SLS}$$

$$\underset{\alpha,\beta}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(z + \alpha \Delta z_1 + \beta \Delta z_2) - \mathcal{T}\|_{\mathsf{F}}^2 \tag{PS}$$

$$\underset{\alpha,\beta,\gamma}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\gamma z + \alpha \Delta z_1 + \beta \Delta z_2) - \mathcal{T}\|_{\mathsf{F}}^2 \tag{SPS}$$

Problem \ Field	$\mathbb{R}$	$\mathbb{C}$
LS	degree $2N$ analytic univariate polynomial	coordinate degree N polyanalytic univariate polynomial
SLS	degree $2N$ analytic univariate rational function	coordinate degree N polyanalytic univariate rational function
PS	total degree $2N$ bivariate polynomial	—
SPS	total degree $2N$ bivariate rational function	—

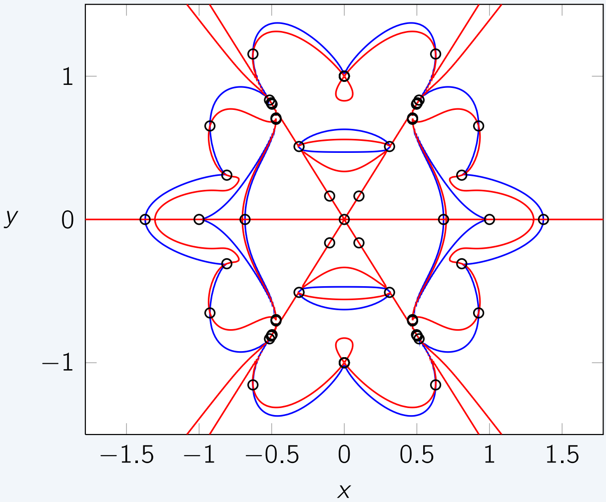
(S)LS-$\mathbb{C}$ and (S)PS-$\mathbb{R}$ are equivalent to solving

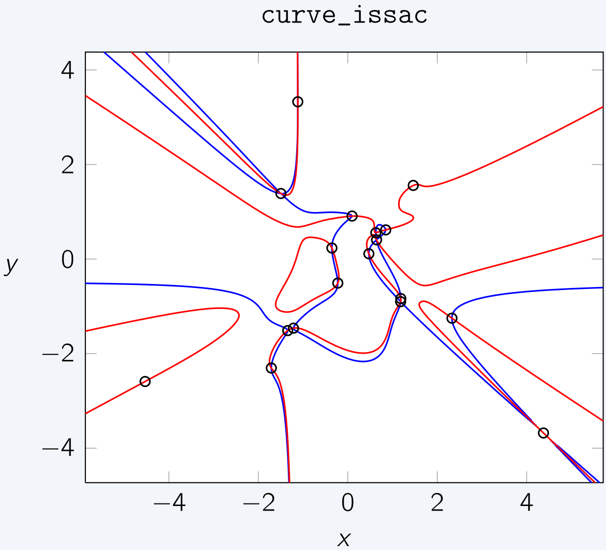
$$\begin{cases} p(x,y) = 0 \\ q(x,y) = 0 \end{cases} \text{where } x, y \in \mathbb{R}$$

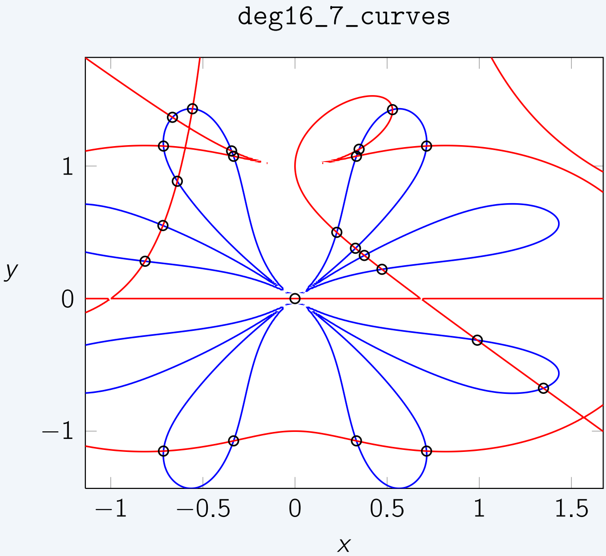
for some polynomials p and q

How? Newton's method, interval methods, semidefinite programming, Gröbner bases, resultants, homotopy continuation, ...

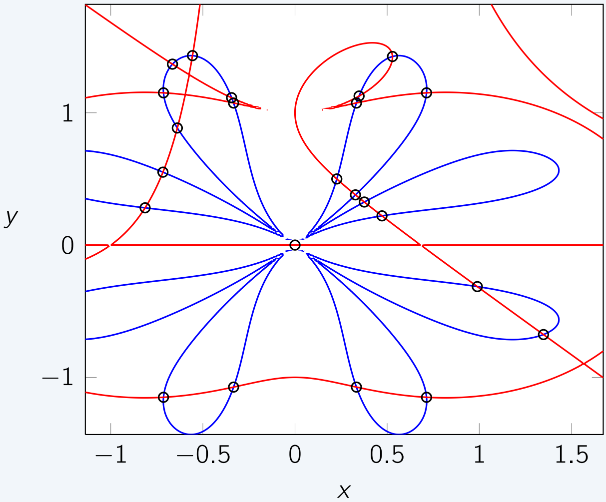
`compact_surf`

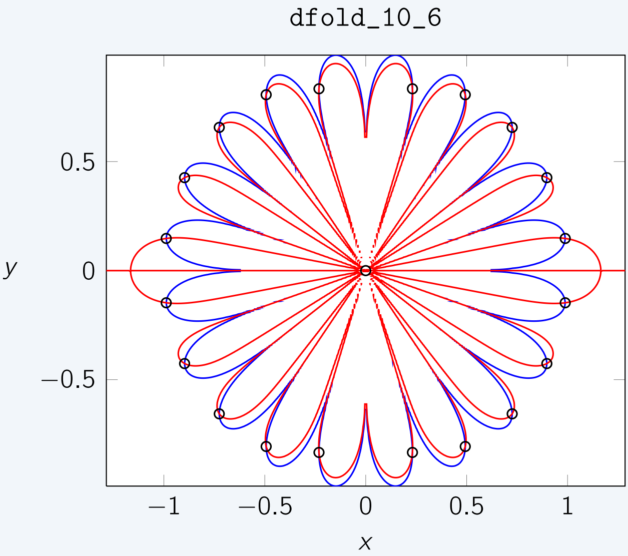


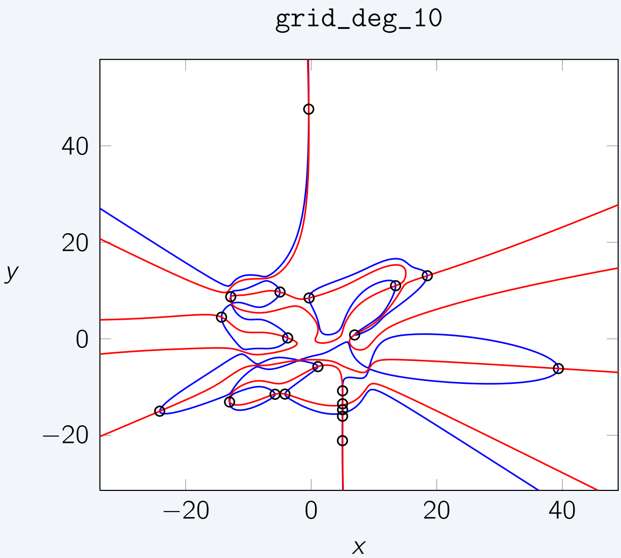

curve_issac
4
2
0
−2
−4
y
−4
−2
0
2
4
x


deg16_7_curves
y
1
0
−1
−1
−0.5
0
0.5
1
1.5
x

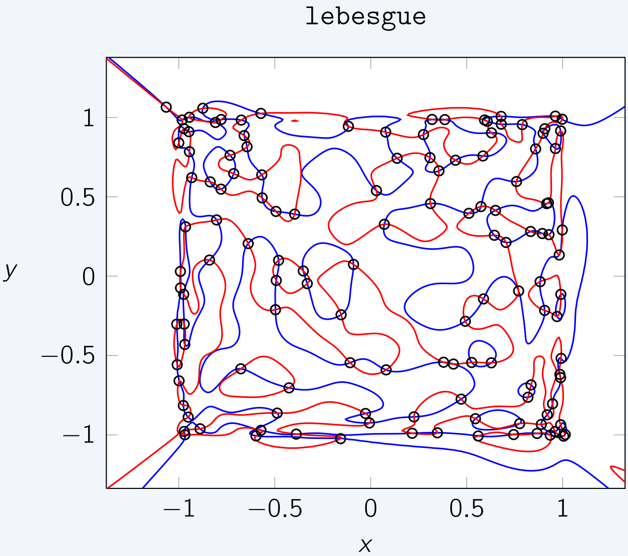
deg16_7_curves



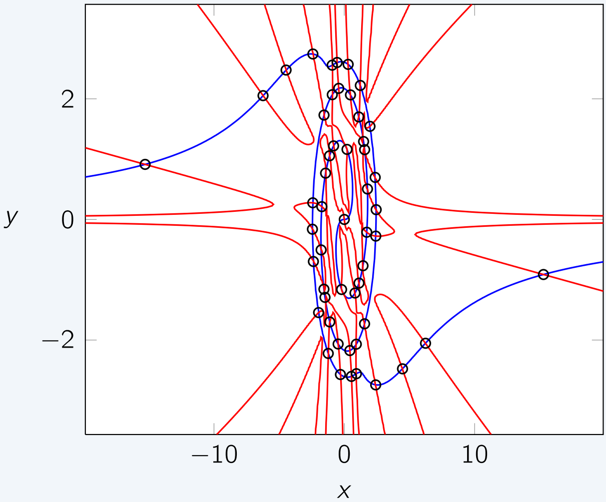

dfold_10_6
0.5
0
−0.5
y
−1
−0.5
0
0.5
1
x



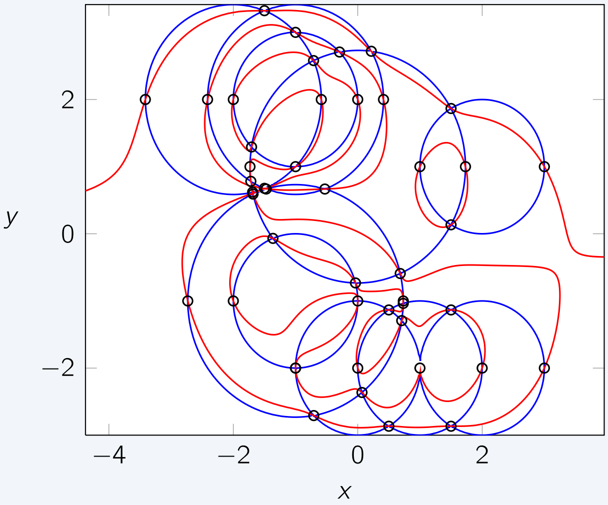
grid_deg_10
40
20
0
−20
y
−20
0
20
40
x

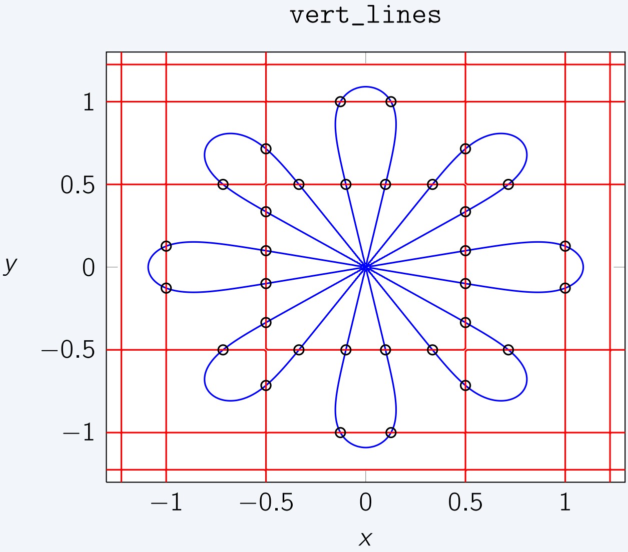

lebesgue
1
0.5
0
−0.5
−1
y
−1
−0.5
0
0.5
1
x

spiral29_24

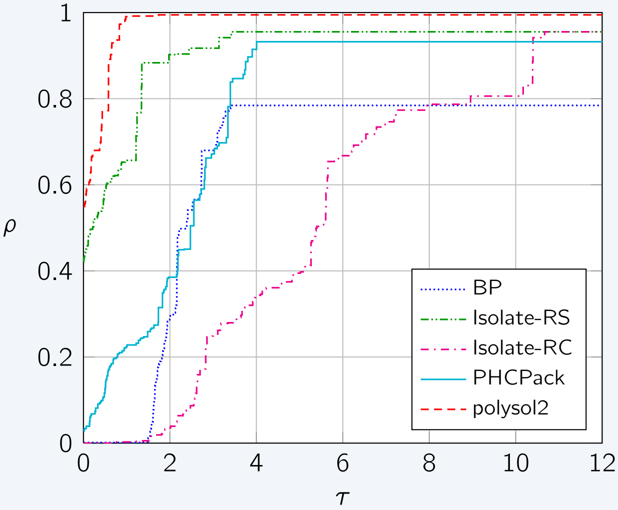


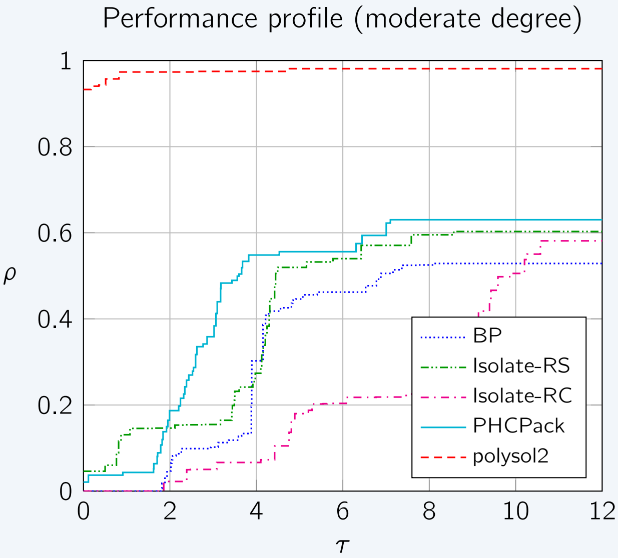
ten_circles




vert_lines
1
0.5
0
−0.5
−1
y
−1
−0.5
0
0.5
1
x

Performance profile (low degree)




Performance profile (moderate degree)
1
0.8
0.6
0.4
0.2
0
ρ
0
2
4
6
8
10
12
τ
BP
Isolate-RS
Isolate-RC
PHCPack
polysol2

Tensorlab v2.0

www.tensorlab.net

Tensorlab v1.0

www.tensorlab.net

A MATLAB toolbox for tensor computations

- Tensor decompositions
  `cpd` `lmlra` `btd`
- Complex optimization
  `minf_lbfgs` `nls_gndl`
- Bivariate polynomial systems
  `polymin2` `polysol2`
- Visualization, rank estimation, statistics, ...
  `voxel3` `rankest` `mlrankest` `cum4`

Tensorlab v2.0

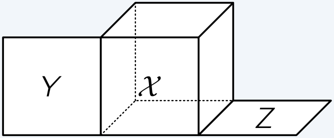
www.tensorlab.net

Major upgrade which brings:

- Full support for sparse and incomplete tensors
- Structured data fusion

  Structured: choose from a large library of constraints to impose on factors (nonnegative, orthogonal, Toeplitz, ...)

  Data fusion: jointly factorize multiple data sets



Example 1: eigenvalue decomposition

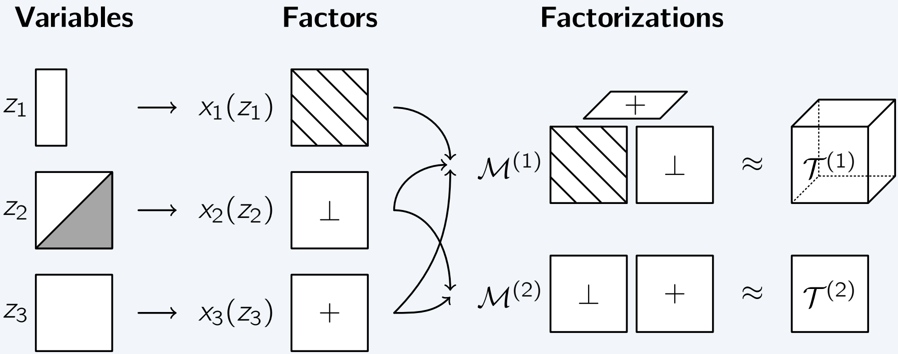
The colleague matrix

$$A = \begin{bmatrix} 0 & 1/2 & & \\ 1 & 0 & 1/2 & \\ & 1/2 & 0 & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

of order n has eigenvalues

$$\lambda_i = \cos\left(\frac{\pi(2i+1)}{2n}\right)$$

for $i = 0, \ldots, n-1$


Variables
Factors
Factorizations
z_1
z_2
z_3
$x_1(z_1)$
$x_2(z_2)$
$x_3(z_3)$
$\mathcal{M}^{(1)}$
$\mathcal{M}^{(2)}$
$\approx$
$\mathcal{T}^{(1)}$
$\mathcal{T}^{(2)}$

Example 1: eigenvalue decomposition

In MATLAB (solve EVD):

```
[V,D] = eig(A);
```

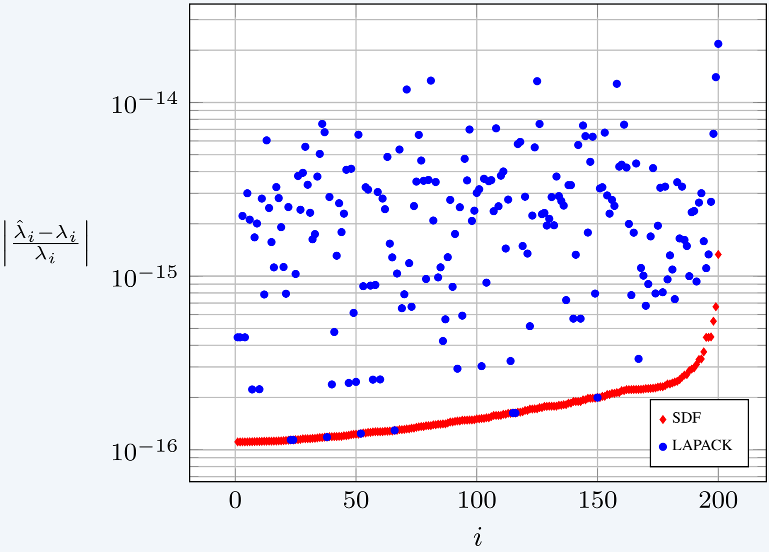
With SDF (define *and* solve EVD):

```
model.variables.v = randn(size(V));
model.variables.d = randn(1,length(D));
```

```
model.factors.V    =  'v';
model.factors.Vinv = {'v',@struct_invtransp};
model.factors.D    =  'd';
```

```
model.factorizations.evd.data = A;
model.factorizations.evd.cpd  = {'V','Vinv','D'};
sol = sdf_nls(model); % sol.factors, sol.variables
```

Example 1: eigenvalue decomposition



Example 2: Netflix \$1M challenge

An incomplete 480k users x 18k movies x 2k timestamps tensor containing 100M integer ratings between 1 and 5 stars

Challenge: predict movie ratings with a RMSE which is 10 % better than Netflix's proprietary Cinematch algorithm

Solution with SDF: model ratings as mean + user bias + movie bias + time bias + low-rank:

$$r_{u,m,t} = \mu + b_u + b_m + b_t + \sum_k a_{u,k} b_{m,k} c_{t,k}$$

Bias vectors are in fact structured rank-1 tensors $\Rightarrow$ model is a structured CPD

Example 2: Netflix \$1M challenge

Model	**RMSE on validation set**
Mean	1.1296
Cinematch	0.9474
Bias + rank-1	0.9447
Bias + rank-2	0.9387
Bias + rank-3	0.9372
Bias + rank-4	0.9326
Bias + rank-5	0.9298
Bias + rank-6	0.9275

We have worked hard so that large data sets such as this 2 GB example can be easily factorized with Tensorlab!

Example 3: InsPyro materials data set

An incomplete tensor in which each dimension represents the concentration of a metal in an alloy (e.g., 9 dimensions)

The tensor's entries are the melting temperatures of an alloy comprising of the selected concentrations

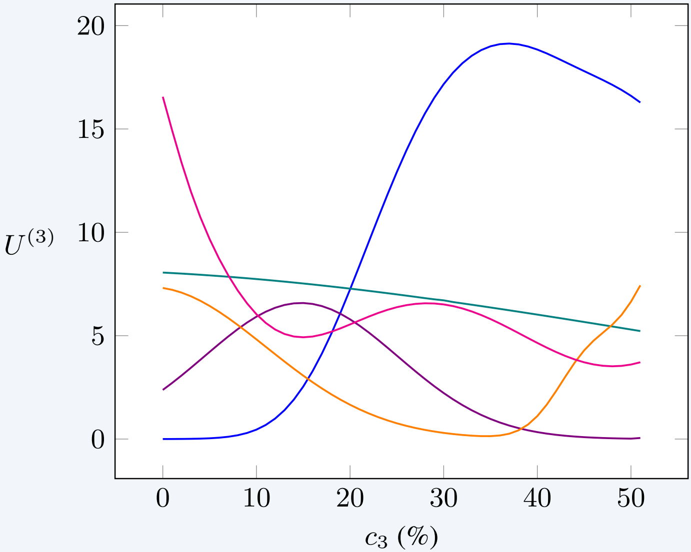
Challenge: predict melting temperatures of different alloys

Solution with SDF: use structured CPD where each factor vector $\boldsymbol{u}_r^{(n)}$ is a sum of RBF kernels

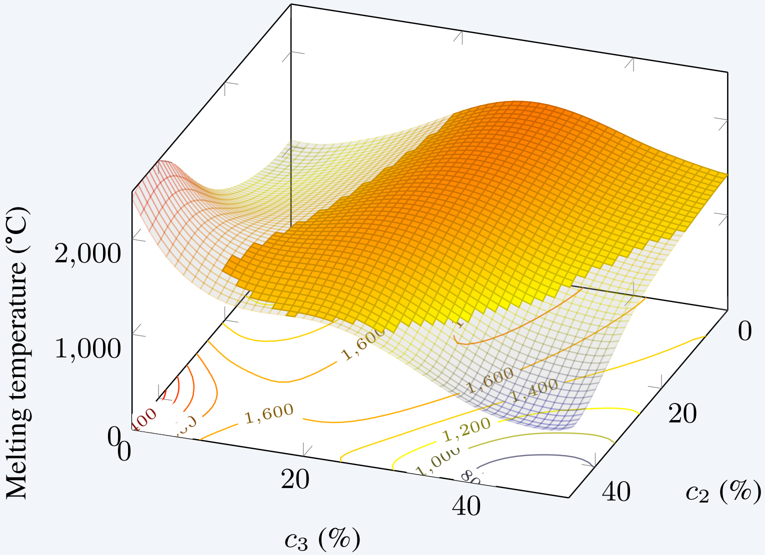
$$u_{b,r}^{(n)} = \sum_{i=1}^{8} a \exp\left(-(t-b)^2/(2c^2)\right)$$

where a b and c are the free parameters in $\boldsymbol{u}_r^{(n)}$

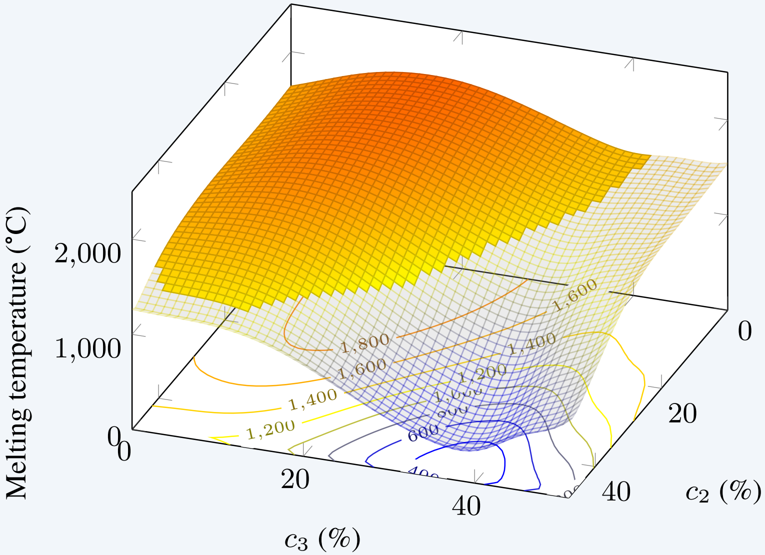
Example 3: InsPyro materials data set



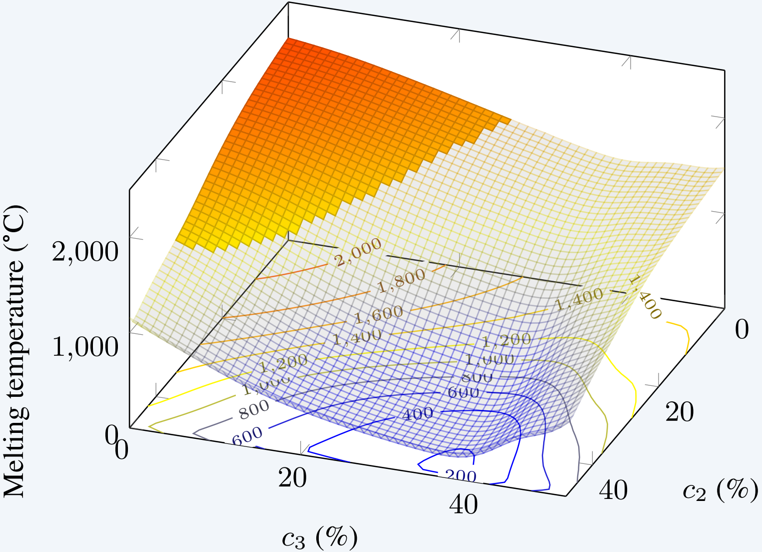
Example 3: InsPyro materials data set



Example 3: InsPyro materials data set



Example 3: InsPyro materials data set



Example 4: GPS data set

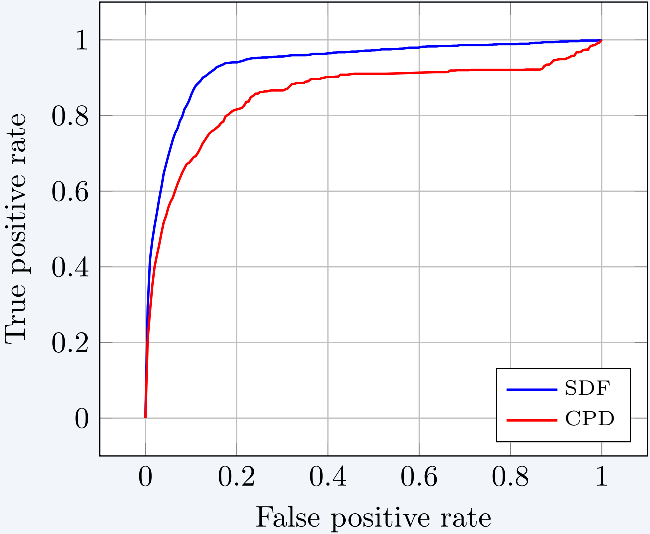
Five coupled data sets: *user-location-activity*, *user-user*, *location-feature*, *activity-activity* and *user-location*

Challenge: predict user participation in activities

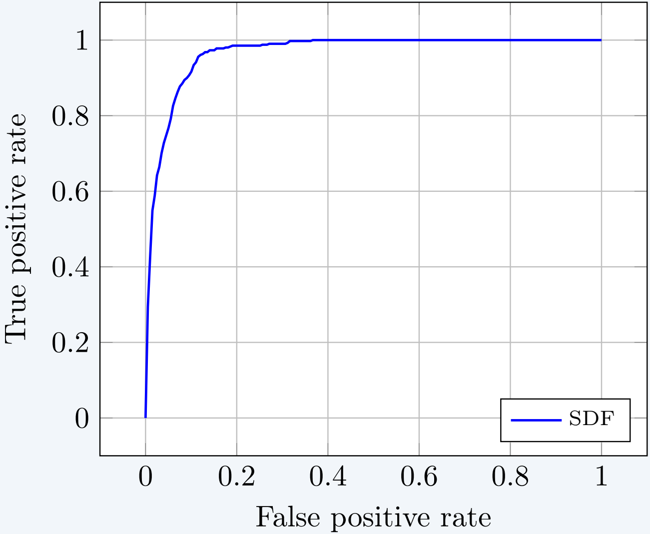
Solution with SDF: compute coupled tensor factorization

$$
\begin{aligned}
\underset{U,L,A,F,\lambda,\mu,\nu}{\text{minimize}} \quad & \frac{\omega_1}{2} \left\| \mathcal{M}^{(1)}(U,L,A) - \mathcal{T}^{(1)} \right\|^2_{\mathcal{W}^{(1)}} \\
& + \frac{\omega_2}{2} \left\| \mathcal{M}^{(2)}(U,U,\lambda) - \mathcal{T}^{(2)} \right\|^2 + \frac{\omega_3}{2} \left\| \mathcal{M}^{(3)}(L,F) - \mathcal{T}^{(3)} \right\|^2 \\
& + \frac{\omega_4}{2} \left\| \mathcal{M}^{(4)}(A,A,\mu) - \mathcal{T}^{(4)} \right\|^2 + \frac{\omega_5}{2} \left\| \mathcal{M}^{(5)}(U,L,\nu) - \mathcal{T}^{(5)} \right\|^2 \\
& + \frac{\omega_6}{2} \left(\|U\|^2 + \|L\|^2 + \|A\|^2 + \|F\|^2 + \|\lambda\|^2 + \|\mu\|^2 + \|\nu\|^2 \right)
\end{aligned}
$$

Example 4: 80% missing entries in *user-location-activity* tensor



Example 4: 50 users missing in *user-location-activity* tensor



Conclusion

- Complex optimization
- Quasi-Newton/NLS vs ALS
- Exact (scaled) line/plane search
- Sets of two bivariate polynomials in real unknowns
- Structured factors: orthogonal, nonnegative, matrix inverse, Toeplitz, Hankel, sums of exponentials, exponentially damped sinusoids, radial basis functions, exponential polynomials, rational functions, . . .
- Coupled decompositions
- www.tensorlab.net