The curse and blessing of dimensionality

Lieven De Lathauwer

Joint work with Nico Vervliet, Martijn Boussé and Otto Debals

August 31, 2018

KU LEUVEN
erc European Research Council
RICHTING MORGEN
Vlaanderen in Actie Pact 2020
Vlaamse overheid
YouReCa
EURASIP
IEEE EMB

Overview

Curse of dimensionality

Algorithms

Variants and applications

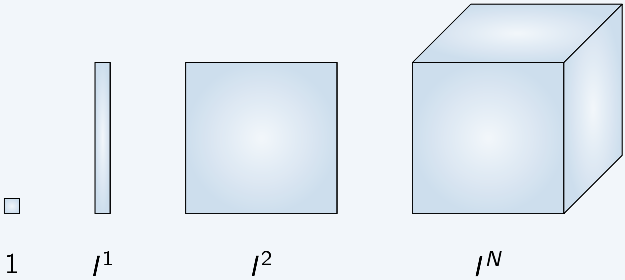
Curse of dimensionality

The curse

Tensor decompositions as a remedy

Immunization by low rank

Tensors



Tensors

- Multidimensional array of numerical values
- General Nth order tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$
- Number of elements: $\mathcal{O}\left(I^N\right)$

Curse of Dimensionality

The problems arising from the exponential increase in memory and computational requirements

Example: # entries in a Nth order tensor of size $100 \times 100 \times \cdots \times 100$ exceeds # atoms in observable universe for $N > 41$

Alleviating or breaking the curse of dimensionality

- Use decompositions
  - Canonical Polyadic Decomposition
  - Low Multilinear Rank Approximation (Tucker, MLSVD)
  - Tensors Trains
  - Hierarchical Tucker
  - Tensor Networks

Alleviating or breaking the curse of dimensionality

- Use decompositions
  - Canonical Polyadic Decomposition
  - Low Multilinear Rank Approximation (Tucker, MLSVD)
  - Tensors Trains
  - Hierarchical Tucker
  - Tensor Networks
- Scientific computing vs signal processing/data analysis

Alleviating or breaking the curse of dimensionality

- Use decompositions
  - Canonical Polyadic Decomposition
  - Low Multilinear Rank Approximation (Tucker, MLSVD)
  - Tensors Trains
  - Hierarchical Tucker
  - Tensor Networks
- Scientific computing vs signal processing/data analysis
- Use incomplete tensors
  - Because we do not have the full tensor
  - Because we do not want the full tensor

[Hackbusch, 2012; Grasedyck et al., 2013; Khoromskij, 2012; Vervliet et al., 2014; Cichocki et al., 2016, 2017]

Curse of dimensionality

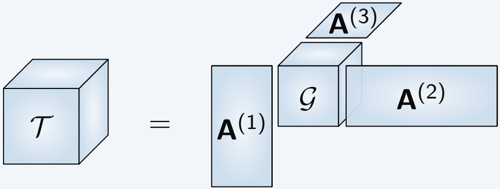
The curse

Tensor decompositions as a remedy

Immunization by low rank

Low Multilinear Rank Approximation

- Multilinear transform of a core tensor



- Mathematically, for a general Nth order tensor $\mathcal{T} \in \mathbb{C}^{I \times \cdots \times I}$

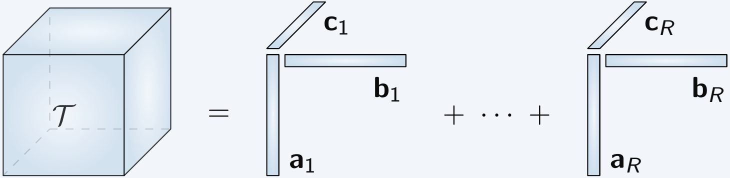
$$\mathcal{T} = \mathcal{G} \cdot_1 \mathbf{A}^{(1)} \cdot_2 \mathbf{A}^{(2)} \cdots \cdot_N \mathbf{A}^{(N)} \triangleq [\![\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]$$

- Number of variables: $\mathcal{O}\left(NIR + R^N\right)$
- Curse not broken, but can be computed via QR/SVD
- Truncation error bound:

$$\left|\left|\mathcal{T} - \hat{\mathcal{T}}_{\text{MLSVD, trunc}}\right|\right|_F^2 \leqslant N \min_{\text{rank}_{\boxplus}(\hat{\mathcal{T}}) \leqslant (R_1, R_2, \ldots, R_N)} \left|\left|\mathcal{T} - \hat{\mathcal{T}}\right|\right|_F^2$$

Canonical polyadic decomposition

- Sum of rank-1 terms



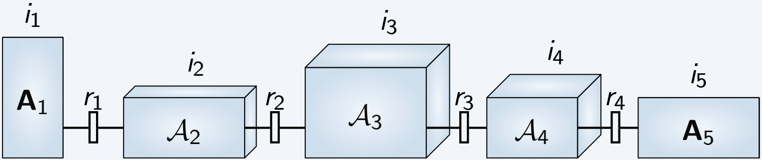
- Mathematically, for a general Nth order tensor $\mathcal{T} \in \mathbb{C}^{I \times \cdots \times I}$

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)} \rrbracket$$

- Number of variables: $\mathcal{O}\left(NIR\right)$
- Curse broken, but possibly ill-conditioned/ill-posed problem

Tensor Trains or Matrix Product States

- Write tensor as a train of lower-order tensors [Oseledets, 2011]



- Mathematically, for a general Nth order tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$

$$t_{i_1 i_2 \cdots i_N} = \sum_{r_1, r_2, \ldots, r_{N-1}} a^{(1)}_{i_1 r_1} a^{(2)}_{r_1 i_2 r_2} a^{(3)}_{r_2 i_3 r_3} \cdots a^{(N-1)}_{r_{N-2} i_{N-1} r_{N-1}} a^{(N)}_{r_{N-1} i_N}$$

- Number of variables: $\mathcal{O}\left(2IR + (N-2)IR^2\right)$
- Curse broken and can be computed via QR/SVD
- Truncation error bound

$$\|\mathcal{T} - \hat{\mathcal{T}}_{\text{TT, trunc}}\|_F^2 \leqslant (N-1) \min_{\text{rank}_{\text{TT}}(\hat{\mathcal{T}}) \leqslant (R_1, R_2, \ldots, R_{N-1})} \|\mathcal{T} - \hat{\mathcal{T}}\|_F^2$$

- hTucker [Hackbusch, 2012]

Curse of dimensionality

The curse

Tensor decompositions as a remedy

Immunization by low rank

Key assumption: low rank

Matrix: decaying singular value spectrum

Power law

Exponential polynomial structure (see further)

Rank-1 terms:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \cdots \otimes \mathbf{u}_r^{(N)}$$

$$\mathbf{T}_{[1,2,\dots;n+1,n+2,\dots]} = (\mathbf{U}^{(1)} \odot \mathbf{U}^{(2)} \odot \cdots) \cdot (\mathbf{U}^{(n+1)} \odot \mathbf{U}^{(n+2)} \odot \cdots)^T$$

$\longrightarrow$ all matrix representations have rank $\leqslant R$

Overview

Curse of dimensionality

Algorithms

Variants and applications

Algorithms for large-scale tensors

Missing entries / partially sampled tensors and CPD

Randomized block sampling for CPD

How to handle large tensors?

- Use incomplete tensors
  - CPWOPT [Acar et al., 2011]
  - CPDI NLS [Vervliet et al., 2014, 2016a]
- Exploit sparsity
  - GigaTensor [Kang et al., 2012]
  - ParCube [Papalexakis et al., 2012]
- Compress the tensor
  - PARACOMP algorithm [Sidiropoulos et al., 2014]
  - Tensor Trains [Oseledets and Tyrtyshnikov, 2010]
- Decompose subtensors and combine results
  - ParCube [Papalexakis et al., 2012]
  - Grid PARAFAC [Phan and Cichocki, 2011]
- Parallel
  - ADMoM [Liavas and Sidiropoulos, 2015]
  - Most of the above

Optimization for CPD

- Optimization problem:

$$\min_{\mathbf{A}^{(1)},\mathbf{A}^{(2)},\ldots,\mathbf{A}^{(N)}} \frac{1}{2}\left\|\mathcal{W} * \left(\mathcal{T} - \left[\!\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}\right]\!\right]\right)\right\|_F^2$$

- Algorithms
  - CPWOPT [Acar et al., 2011]
    Nonlinear Conjugate Gradients
  - INDAFAC [Tomasi and Bro, 2005]
    Gauss–Newton
  - CPD/SDF [Sorber et al., 2015]
    Quasi-Newton and (approximate) inexact Gauss–Newton
    Tensorlab: `cpd_nls`, `sdf_nls`
  - CPD(L)I [Vervliet et al., 2016a,d]
    Inexact Gauss–Newton with possible linear constraints
    Tensorlab: `cpd_nls` with `UseCPDI` option, `cpdli_nls`
- Samples investigated: N_{samples}

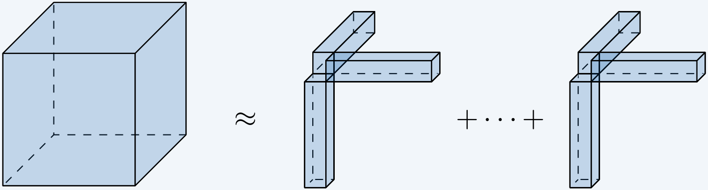
Algorithms for large-scale tensors

Missing entries / partially sampled tensors and CPD

Randomized block sampling for CPD

Randomized block sampling CPD: idea

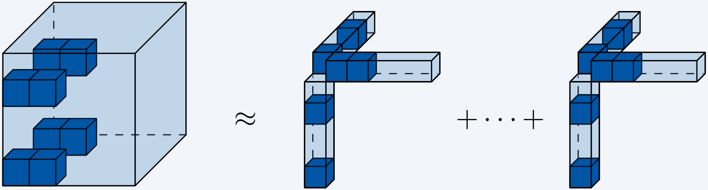
stochastic descent with step restriction



[Vervliet and De Lathauwer, 2016]

Randomized block sampling CPD: idea

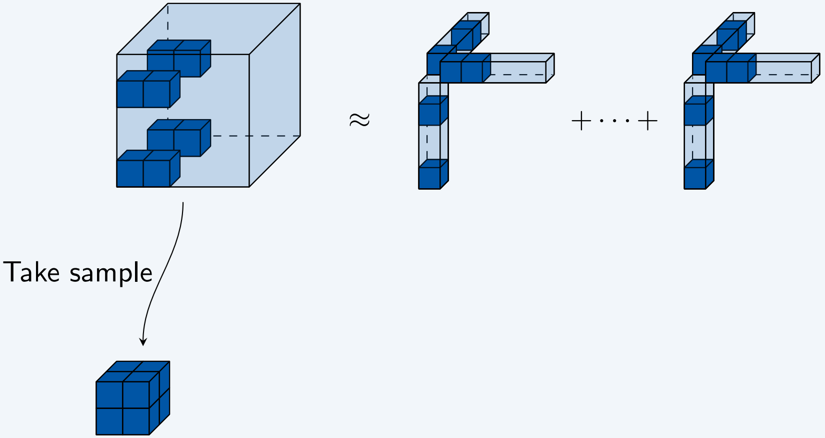
stochastic descent with step restriction



[Vervliet and De Lathauwer, 2016]

Randomized block sampling CPD: idea

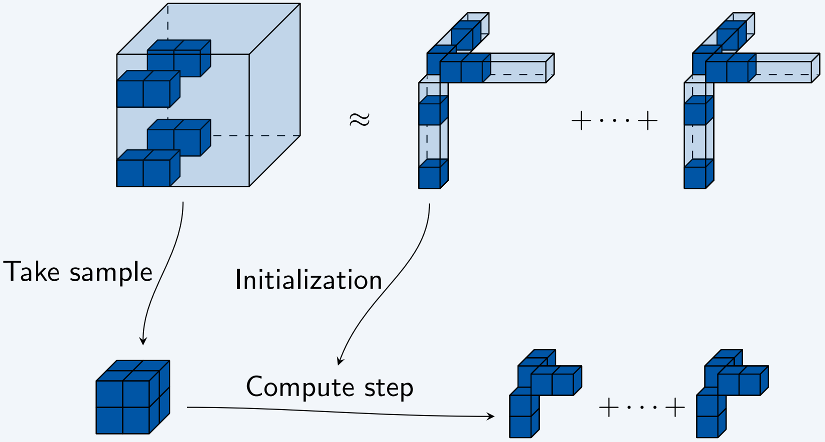
stochastic descent with step restriction



[Vervliet and De Lathauwer, 2016]

Randomized block sampling CPD: idea

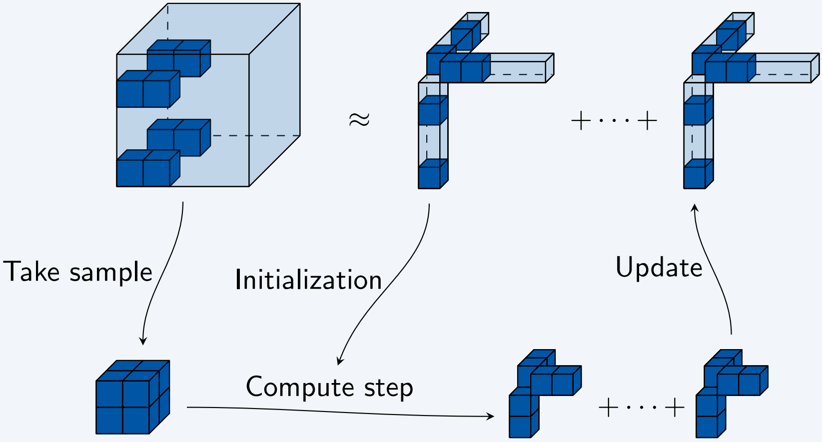
stochastic descent with step restriction



[Vervliet and De Lathauwer, 2016]

Randomized block sampling CPD: idea

stochastic descent with step restriction



[Vervliet and De Lathauwer, 2016]

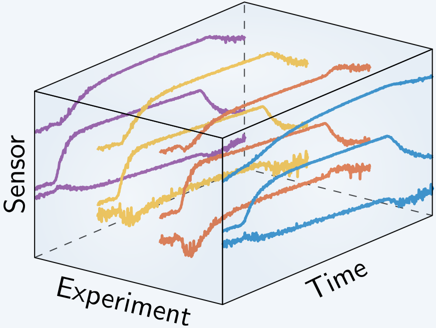
Detection of hazardous gasses using e-noses

Classify 900 experiments containing 72 time series with **26 000** samples each, totaling 12.5 GB of data.

[Vervliet and De Lathauwer, 2016]

Classify hazardous gasses

Does the sample contain CO, acetaldehyde or ammonia?

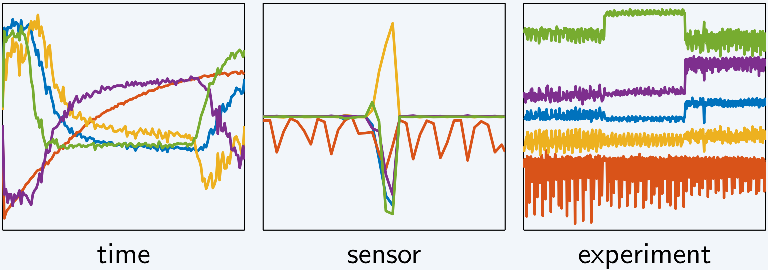


Strategy: classify using coefficients of spatiotemporal patterns.

$26\,000 \times 72 \times 900$ $100 \times 36 \times 100$ $R = 5$ Unknown

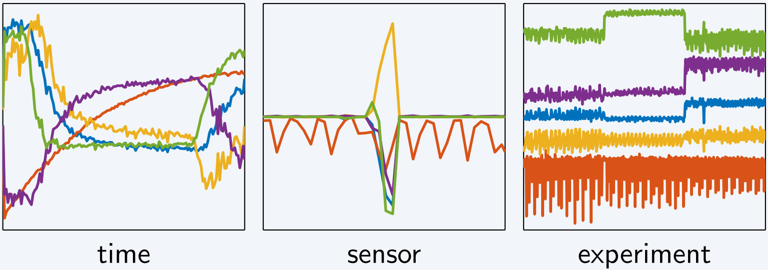
Results

- Resulting factor matrices



Results

- Resulting factor matrices



- Performance after clustering

	Iterations	Time (s)	Error (%)
No restriction	3000	60	5.0
Restriction	9000	170	0.3–0.8

Overview

Curse of dimensionality

Algorithms

Variants and applications

Variants and applications

Compression as preprocessing in the computation of unconstrained and constrained decompositions

Thermodynamic data and curse of dimensionality

No tensor? Quantization and blessing of dimensionality

Exploiting low multilinear rank for tensor decompositions

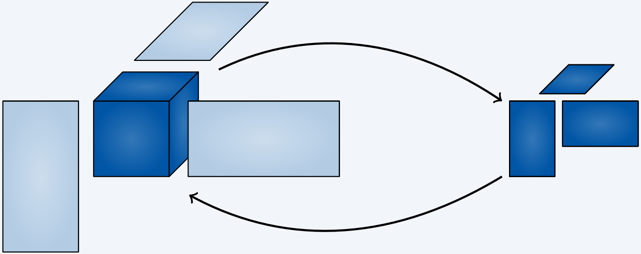
Strategy without constraints:

1 | Compress tensor, e.g, using (randomized) MLSVD [Vervliet et al., 2016c]

2 | Compute CPD of core tensor

3 | Expand CPD using factor matrices of compression

4 | Refine result if necessary



Orthogonal factor matrices preserve length and distance in compression

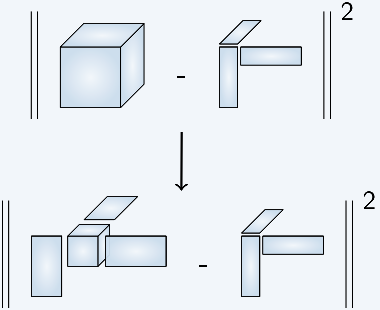
Exploiting low multilinear rank for tensor decompositions

Strategy with constraints [Vervliet et al., 2016c]:

1 | Compute LMLRA

2 | Decompose while exploiting structure

Core operations like norms, inner products and `mtkrprod` exploit structure of the tensor



Many combinations of structures and decompositions possible

Exploiting efficient representations in tensor decompositions

Tensorlab can compute tensors given in an efficient format
CPD, LMLRA, Tensor Train, Hankel, Löwner, ...

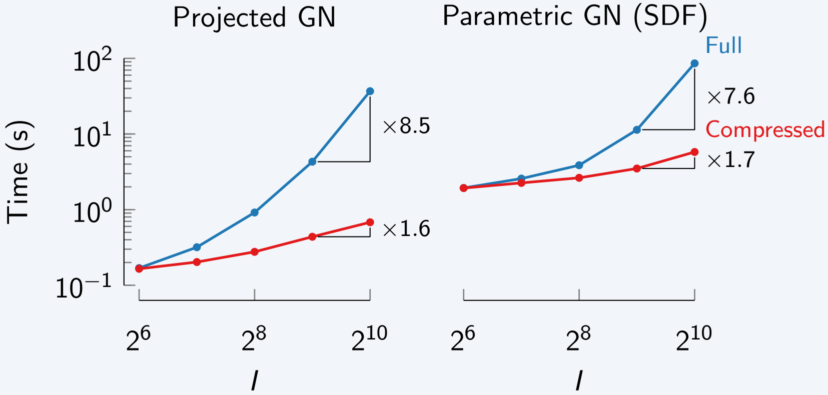
using possibly coupled and/or symmetric decompositions
CPD, LL1, LMLRA, BTD

with possible constraints
nonnegativity, Hankel, Vandermonde, polynomial, orthogonal, ...

Example: compute a nonnegative rank-5 CPD of a $500 \times 500 \times 500$ tensor after randomized MLSVD compression using `mlsvd_rsi`

Nonnegative CPD using MLSVD compression

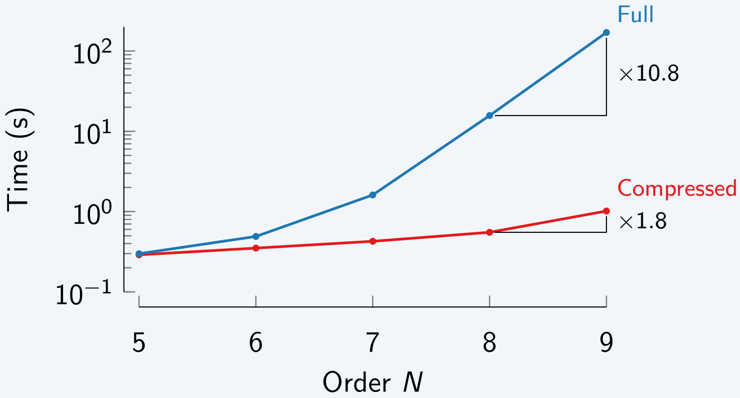
Compute nonnegative rank-10 CPD of $I \times I \times I$ tensor with SNR 20 dB:



[Vervliet et al., 2016b]

Nonnegative CPD using TT compression

Compute nonnegative rank-5 CPD of Nth-order $10 \times \cdots \times 10$ tensor with SNR 20 dB:



[Vervliet et al., 2016b]

Variants and applications

Compression as preprocessing in the computation of unconstrained and constrained decompositions

Thermodynamic data and curse of dimensionality

No tensor? Quantization and blessing of dimensionality

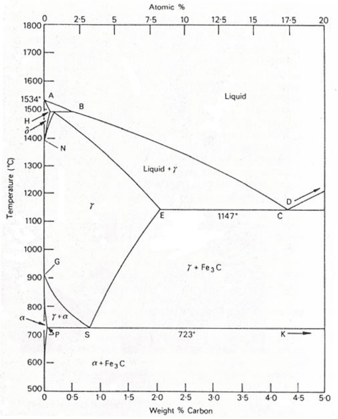
Modeling multiway thermodynamic data

Modes: fraction atom/molecule n in a multi-component material
Value: Gibbs free energy, chemical potential, melting temperature
(e.g., computed using thermodynamic software)

Modeling multiway thermodynamic data

Modes: fraction atom/molecule *n* in a multi-component material
Value: Gibbs free energy, chemical potential, melting temperature (e.g., computed using thermodynamic software)

First-order example



Modeling multiway thermodynamic data

Modes: fraction atom/molecule n in a multi-component material
Value: Gibbs free energy, chemical potential, melting temperature (e.g., computed using thermodynamic software)

Second-order example

- Alloy with c_1 % iron, c_2 % carbon and $100 - c_1 - c_2$ % nickel
- Discretize c_1 and c_2 in 100 steps
- Grid of size 100^2

[Vervliet et al., 2014]

Multiway dataset

- # constituent materials: 10 (thus $N = 9$)
- Size: $100 \times 100 \times \cdots \times 100 \approx 10^{18}$ elements
- # Samples: 130 000 of which 30 000 are validation samples
- Model:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)}$$

Multiway dataset

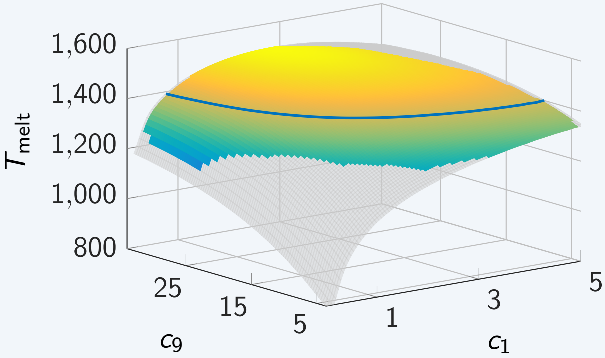
- # constituent materials: 10 (thus $N = 9$)
- Size: $100 \times 100 \times \cdots \times 100 \approx 10^{18}$ elements
- # Samples: 130 000 of which 30 000 are validation samples
- Model:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)}$$

Algorithm

- Tensorlab 3.0 with `cpd_nls` and `UseCPDI` option [Vervliet et al., 2014, 2016d]
- Initialization: optimally scaled best-out-of-five strategy

Visualization



Fitting the model

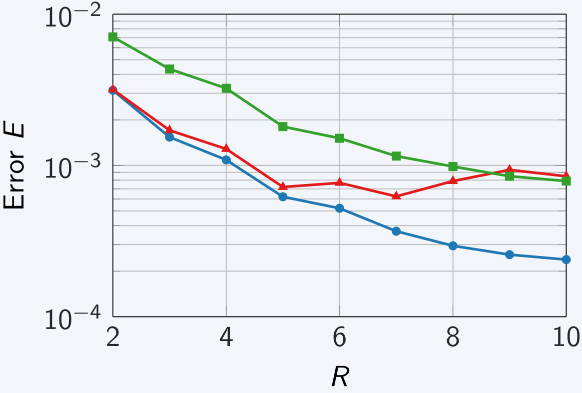


Figure: Errors on training E_{tr} (—●—) and validation E_{val} (—▲—) set and the 99% quantile error E_{quant} (—■—) for different CPDs. The computation time for each model is indicated by (—◆—) on the right y-axis.

Fitting the model

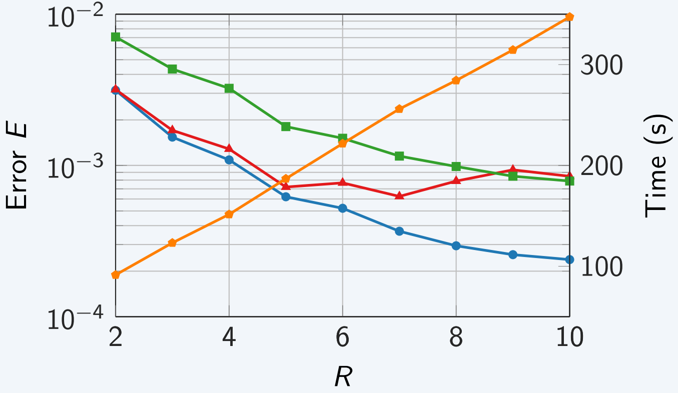
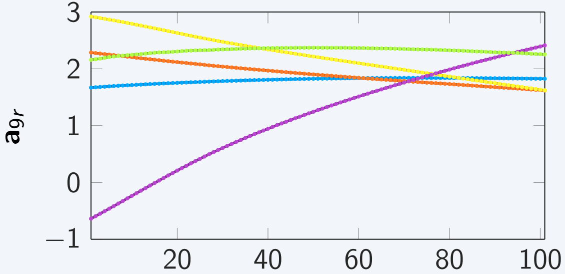
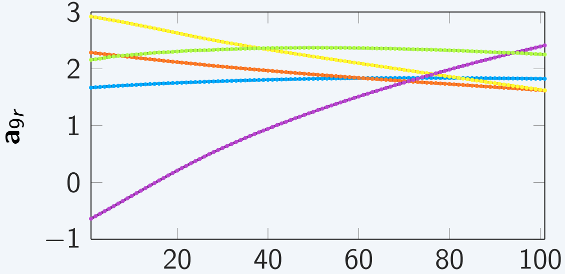


Figure: Errors on training E_{tr} (—●—) and validation E_{val} (—▲—) set and the 99% quantile error E_{quant} (—■—) for different CPDs. The computation time for each model is indicated by (—⬟—) on the right y-axis.

From a discrete model. . .



From a discrete model...



to a continuous (e.g., polynomial) model

$$t_{i_1\cdots i_9} \approx f(c_1, \dots, c_N) = \sum_{r=1}^{R} \prod_{n=1} a_r^{(n)}(c_n),$$

Advantage: allows interpolation, derivation and integration, parameter reduction, ...

Recap

- From a ninth order tensor with 10^{18} elements...

Recap

- From a ninth order tensor with 10^{18} elements...
- we took 100 000 samples...

Recap

- From a ninth order tensor with 10^{18} elements...
- we took 100 000 samples...
- to get a rank-5 model with $5 \times 9 \times 100 = 4\,500$ parameters...

Recap

- From a ninth order tensor with 10^{18} elements...
- we took 100 000 samples...
- to get a rank-5 model with $5 \times 9 \times 100 = 4\,500$ parameters...
- to get a continuous model with $\mathcal{O}(100)$ parameters...

Recap

- From a ninth order tensor with 10^{18} elements...
- we took 100 000 samples...
- to get a rank-5 model with $5 \times 9 \times 100 = 4\,500$ parameters...
- to get a continuous model with $\mathcal{O}(100)$ parameters...
- in 3 min

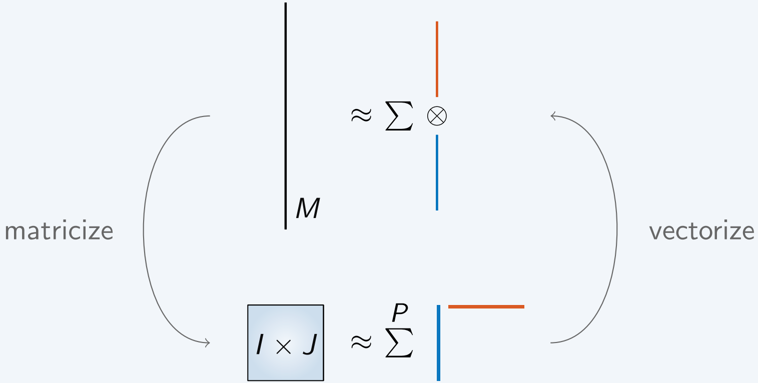
Variants and applications

Compression as preprocessing in the computation of unconstrained and constrained decompositions

Thermodynamic data and curse of dimensionality

No tensor? Quantization and blessing of dimensionality

Low-rank matrices can be used as compact models for large-scale vectors



$$M = IJ \to P(I + J)$$

[Khoromskij, 2012]

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

$$\mathbf{f} = \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{bmatrix}$$

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

$$\mathbf{f} = \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{bmatrix} \longrightarrow \mathbf{R} = \begin{pmatrix} 1 & z^3 \\ z & z^4 \\ z^2 & z^5 \end{pmatrix}$$

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

$$\mathbf{f} = \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{bmatrix} \longrightarrow \mathbf{R} = \begin{pmatrix} 1 & z^3 \\ z & z^4 \\ z^2 & z^5 \end{pmatrix} = \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z^3 \end{pmatrix}$$

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

$$\mathbf{f} = \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{bmatrix} \longrightarrow \mathbf{R} = \begin{pmatrix} 1 & z^3 \\ z & z^4 \\ z^2 & z^5 \end{pmatrix} = \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z^3 \end{pmatrix}$$

R can be interpreted as a *compact* form of the Hankel matrix **H**

$$\mathbf{H} = \begin{pmatrix} 1 & z & z^2 & z^3 \\ z & z^2 & z^3 & z^4 \\ z^2 & z^3 & z^4 & z^5 \end{pmatrix}$$

The approach holds exactly for (exponential) polynomials

Exponential polynomials = sum and/or products of exponentials, sinusoids and/or polynomials

$$\mathbf{f} = \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \\ z^4 \\ z^5 \end{bmatrix} \longrightarrow \mathbf{R} = \begin{pmatrix} 1 & z^3 \\ z & z^4 \\ z^2 & z^5 \end{pmatrix} = \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z^3 \end{pmatrix}$$

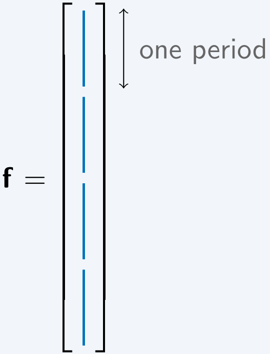
R can be interpreted as a *compact* form of the Hankel matrix **H**

$$\mathbf{H} = \begin{pmatrix} 1 & z & z^2 & z^3 \\ z & z^2 & z^3 & z^4 \\ z^2 & z^3 & z^4 & z^5 \end{pmatrix} = \begin{pmatrix} 1 \\ z \\ z^2 \end{pmatrix} \begin{pmatrix} 1 & z & z^2 & z^3 \end{pmatrix}$$

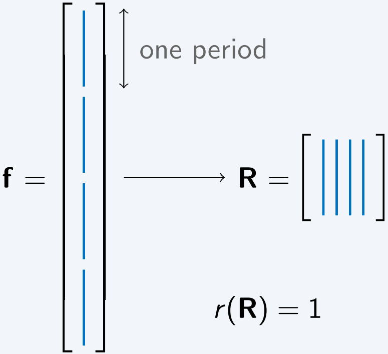
$f(t)$	$r(\mathbf{H})$	$f(t)$	$r(\mathbf{H})$
az^t	1	$\sum_{r=1}^{R} a_r z_r^t$	R
$a\sin(bt)$ $a\cos(bt)$	2	$\sum_{r=1}^{R} a_r \sin(b_r t)$	$2R$
$az^t \sin(bt)$	2	$\sum_{r=1}^{R} a_r z_r^t \sin(b_r t)$	$2R$
$p(t) = \sum_{q=0}^{Q} a_q t^q$	$Q+1$	$\sum_{r=1}^{R} p_r(t)$	$\sum_{r=1}^{R} Q_r + R$
$p(t)z^t$	$Q+1$	$\sum_{r=1}^{R} p_r(t) z_r^t$	$\sum_{r=1}^{R} Q_r + R$

[Boussé et al., 2017]

Periodic signals can be reshaped into low-rank matrices



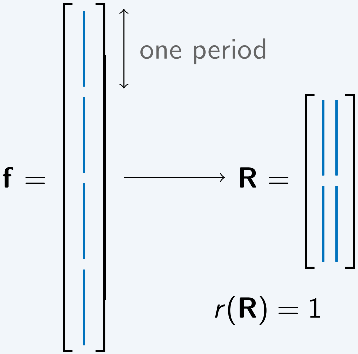
Periodic signals can be reshaped into low-rank matrices



Regardless of the type of signal, e.g., discontinuities are allowed.

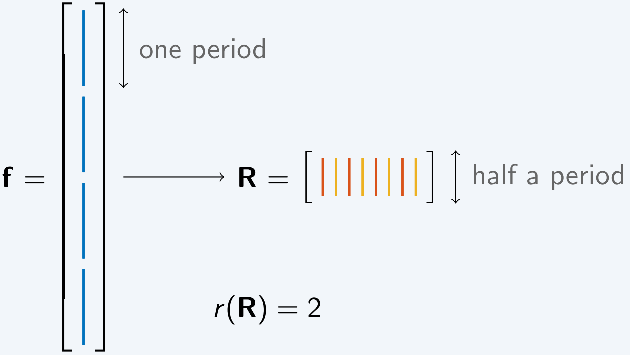
KU LEUVEN

Periodic signals can be reshaped into low-rank matrices



Regardless of the type of signal, e.g., discontinuities are allowed.

Periodic signals can be reshaped into low-rank matrices



Regardless of the type of signal, e.g., discontinuities are allowed.

The approach also works well for more general compressible functions

$$\epsilon = \|\mathbf{f} - \text{vec}(\tilde{\mathbf{R}})\|_\mathsf{F}^2$$

Underlying function $f(t)$

Low-rank approximation of $\mathbf{R} = \text{reshape}(\mathbf{f})$

[Grasedyck et al., 2013; Boussé et al., 2017]

The approach also works well for more general compressible functions

$$\epsilon = \|\mathbf{f} - \mathrm{vec}(\tilde{\mathbf{R}})\|_\mathsf{F}^2$$

Underlying function $f(t)$

Low-rank approximation of $\mathbf{R} = \mathrm{reshape}(\mathbf{f})$

Functions with rapidly converging Taylor series admit an approximate low-rank model

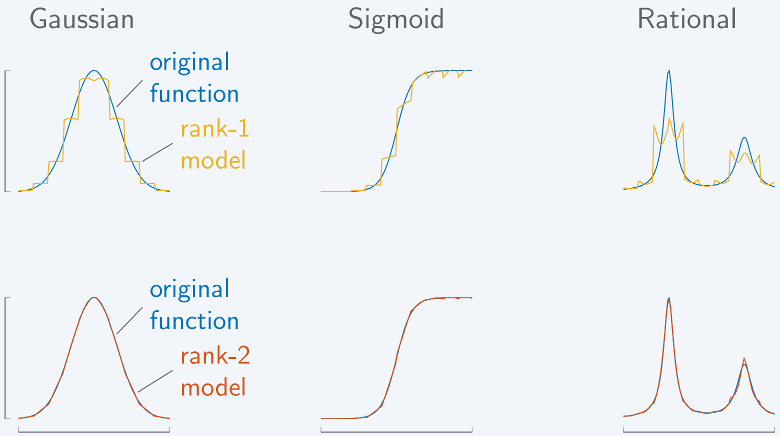
$$|f(t) - p(t)| \leq \epsilon_{\mathsf{Taylor}}$$

Taylor polynomial

Error bound!

[Grasedyck et al., 2013; Boussé et al., 2017]

The singular values of **R** often decay fast, hence, **f** often admits a good representation for low rank values.



Conclusion

Tensor problems are often large-scale

Transfer of know-how from scientific computing in high dimensions to big data analysis

Alleviate/break curse of dimensionality

- using decompositions for analysis, compression, ...
- by computations using randomization, incompleteness, efficient representations, ...

Dimensionality is also a blessing

- segmentation/quantization

The curse and blessing of dimensionality

Lieven De Lathauwer

Joint work with Nico Vervliet, Martijn Boussé and Otto Debals

August 31, 2018

References I

E. Acar, D.M. Dunlavy, T.G. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, Mar 2011.

M. Boussé, O. Debals, and L. De Lathauwer. A tensor-based method for large-scale blind source separation using segmentation. *IEEE Transactions on Signal Processing*, Jan. 2017.

A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and A.-H. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, Mar 2015.

References II

A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D.P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 2016.

A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. Oseledets, M. Sugiyama, D.P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 2017.

L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. Feb. 2013.

W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*, volume 42. Springer Science & Business Media, 2012.

References III

U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos. GigaTensor: scaling tensor analysis up by 100 times — algorithms and discoveries. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

B.N. Khoromskij. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems*, 2012.

A. Liavas and N. Sidiropoulos. Parallel algorithms for constrained tensor factorization via the alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 2015.

I.V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 2011.

References IV

I.V. Oseledets and E.E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 2010.

E. Papalexakis, C. Faloutsos, and N. Sidiropoulos. ParCube: Sparse parallelizable tensor decompositions. In P.A. Flach, T. De Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7523 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012.

A.-H. Phan and A. Cichocki. PARAFAC algorithms for large-scale problems. *Neurocomputing*, 2011.

N. Sidiropoulos, E. Papalexakis, and C. Faloutsos. Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition. *IEEE Signal Processing Magazine*, 2014.

References V

N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, July 2017.

L. Sorber, M. Van Barel, and L. De Lathauwer. Structured data fusion. *IEEE Journal of Selected Topics in Signal Processing*, June 2015.

G. Tomasi and R. Bro. Parafac and missing values. *Chemometrics and Intelligent Laboratory Systems*, Feb 2005.

N. Vervliet and L. De Lathauwer. A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors. *IEEE Journal of Selected Topics in Signal Processing*, Mar 2016.

References VI

N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Processing Magazine*, Sep 2014.

N. Vervliet, O. Debals, and L. De Lathauwer. Canonical polyadic decomposition of incomplete tensors with linearly constrained factors. Technical Report 16–172, ESAT-STADIUS, KU Leuven, Belgium, 2016a.

N. Vervliet, O. Debals, and L. De Lathauwer. Exploiting efficient data representations in tensor decompositions. Technical Report 16–174, ESAT-STADIUS, KU Leuven, Belgium, 2016b.

References VII

N. Vervliet, O. Debals, and L. De Lathauwer. Tensorlab 3.0 — numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In *2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016c.

N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. Tensorlab 3.0, Mar 2016d. Available online at `http://www.tensorlab.net`.