A randomized block sampling approach to the canonical polyadic decomposition of large-scale tensors

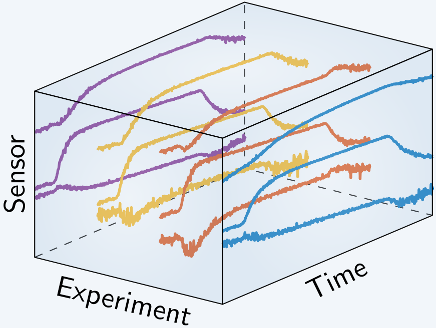
Nico Vervliet

Joint work with Lieven De Lathauwer

SIAM AN17, July 13, 2017

Classification of hazardous gasses using e-noses

Classify 900 experiments containing 72 time series with 26 000 samples each.



Overview

Decomposing large-scale tensors

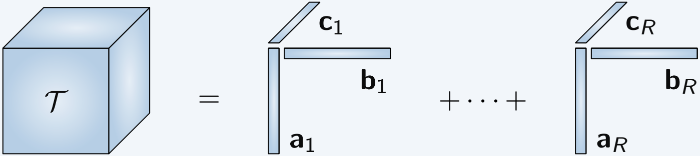
Randomized block sampling

Experimental results

Chemo-sensing application

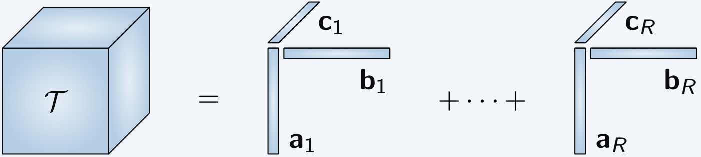
Canonical polyadic decomposition

- Sum of R rank-1 terms



Canonical polyadic decomposition

- Sum of R rank-1 terms



- Mathematically, for a general Nth order tensor $\mathcal{T}$

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \cdots \otimes \mathbf{a}_r^{(N)}$$
$$= \left[\!\left[\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)} \right]\!\right]$$

Computing a CPD

- Optimization problem:

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

Computing a CPD

- Optimization problem:

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

- Algorithms
  - Alternating least squares
  - CPOPT [Acar et al. 2011a]
  - (Damped) Gauss–Newton [Phan et al. 2013]
  - (Inexact) nonlinear least squares [Sorber et al. 2013]

Curse of dimensionality

- Suppose Nth order $\mathcal{T} \in \mathbb{C}^{I \times I \times \cdots \times I}$, then
- number of entries: I^N
- memory and time complexity: $\mathcal{O}\left(I^N\right)$

Curse of dimensionality

- Suppose Nth order $\mathcal{T} \in \mathbb{C}^{I \times I \times \cdots \times I}$, then
- number of entries: I^N
- memory and time complexity: $\mathcal{O}\left(I^N\right)$
- number of variables: NIR

Curse of dimensionality

- Suppose Nth order $\mathcal{T} \in \mathbb{C}^{I \times I \times \cdots \times I}$, then
- number of entries: I^N
- memory and time complexity: $\mathcal{O}\left(I^N\right)$
- number of variables: NIR

Example [Vervliet et al. 2014]

Ninth-order tensor with $I = 100$ and rank $R = 5$:

- number of entries: 10^{18}
- number of variables: 4500

How to handle large tensors?

- Use incomplete tensors
  Acar et al. 2011b; Vervliet et al. 2014; Vervliet et al. 2016a
- Exploit sparsity
  Kang et al. 2012; Papalexakis et al. 2012; Bader and Kolda 2007
- Compress the tensor
  Sidiropoulos et al. 2014; Oseledets and Tyrtyshnikov 2010; Vervliet et al. 2016b
- Decompose subtensors and combine results
  Papalexakis et al. 2012; Phan and Cichocki 2011
- Parallel
  Liavas and Sidiropoulos 2015 + many of the above

Overview

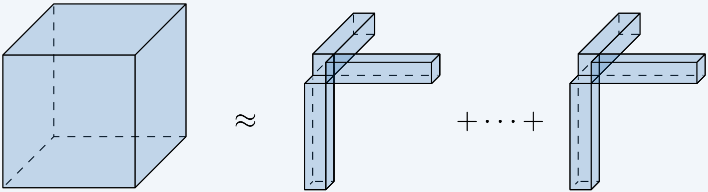
Decomposing large-scale tensors

Randomized block sampling

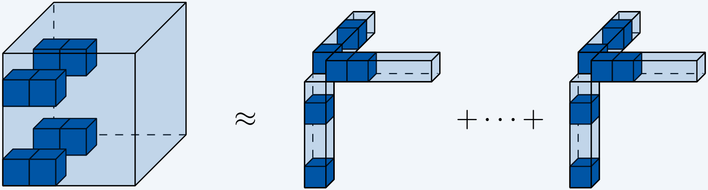
Experimental results

Chemo-sensing application

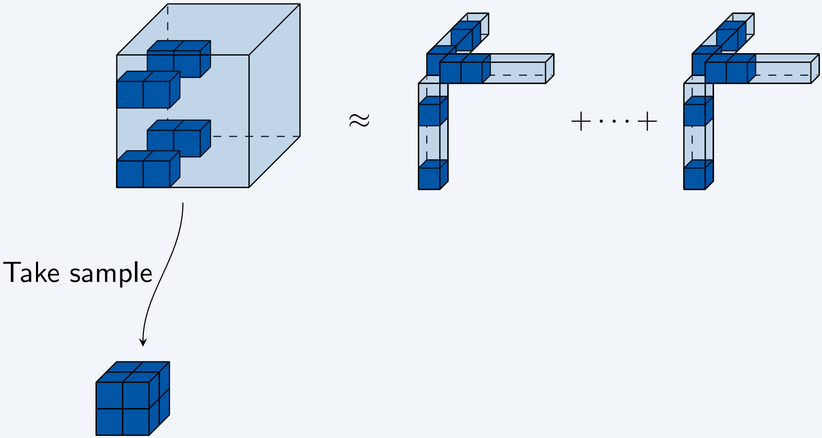
Randomized block sampling CPD: idea



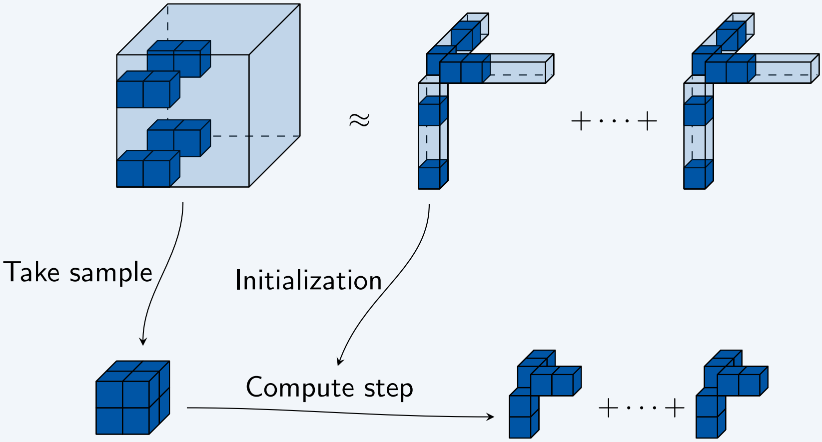
Randomized block sampling CPD: idea



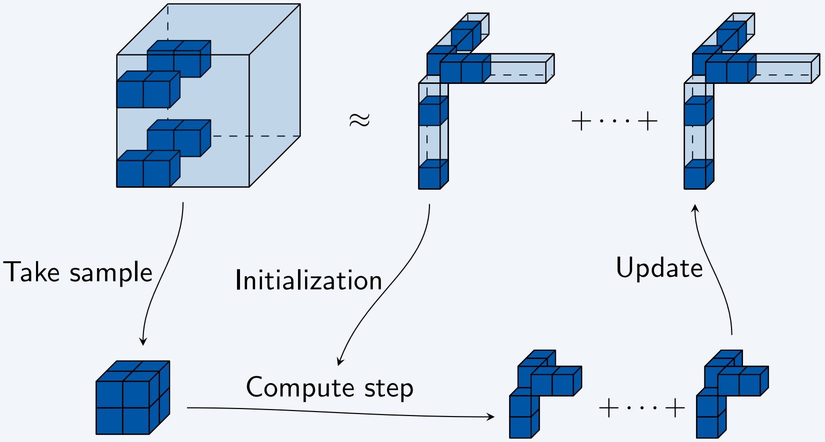
Randomized block sampling CPD: idea



Randomized block sampling CPD: idea



Randomized block sampling CPD: idea



Randomized block sampling CPD: algorithm

input : Data $\mathcal{T}$ and initial guess $\mathbf{A}^{(n)}$, $n = 1, ..., N$
output: $\mathbf{A}^{(n)}$, $n = 1, ..., N$ such that $\mathcal{T} \approx [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$
while $k < K$ *and not converged* **do**

Create sample $\mathcal{T}_s$ and corresponding $\mathbf{A}_s^{(n)}$, $n = 1, \ldots, N$

Let $\bar{\mathbf{A}}_s^{(n)}$ be the result of 1 iteration in a restricted CPD algorithm on $\mathcal{T}_s$ with initial guess $\mathbf{A}_s^{(n)}$, $n = 1, ..., N$ and restriction Δ

Update the *affected* variables $\mathbf{A}^{(n)}$ using $\bar{\mathbf{A}}_s^{(n)}$, $n = 1, ..., N$

$k \leftarrow k + 1$

Randomized block sampling CPD: algorithm

input : Data $\mathcal{T}$ and initial guess $\mathbf{A}^{(n)}$, $n = 1, ..., N$
output: $\mathbf{A}^{(n)}$, $n = 1, ..., N$ such that $\mathcal{T} \approx [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$
while $k < K$ *and not converged* **do**

Create sample $\mathcal{T}_s$ and corresponding $\mathbf{A}_s^{(n)}$, $n = 1, \ldots, N$

Let $\bar{\mathbf{A}}_s^{(n)}$ be the result of 1 iteration in a restricted CPD algorithm on $\mathcal{T}_s$ with initial guess $\mathbf{A}_s^{(n)}$, $n = 1, ..., N$ and restriction Δ

Update the *affected* variables $\mathbf{A}^{(n)}$ using $\bar{\mathbf{A}}_s^{(n)}$, $n = 1, ..., N$

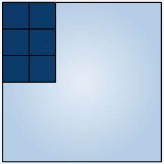
$k \leftarrow k + 1$

Ingredient 1: randomized block sampling

For a 6×6 tensor and block size 3×2:

$\mathcal{I}_1 = \{3, 1, 2, 6, 5, 4\}$

$\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$

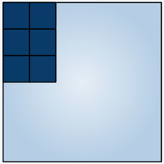


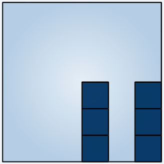
Ingredient 1: randomized block sampling

For a 6×6 tensor and block size 3×2:

$\mathcal{I}_1 = \{3, 1, 2, 6, 5, 4\}$ $\mathcal{I}_1 = \{3, 1, 2, 6, 5, 4\}$

$\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$ $\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$

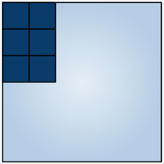




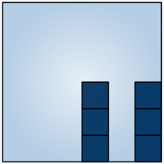
Ingredient 1: randomized block sampling

For a 6×6 tensor and block size 3×2:

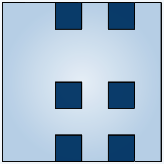
$\mathcal{I}_1 = \{3, 1, 2, 6, 5, 4\}$
$\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$



$\mathcal{I}_1 = \{3, 1, 2, 6, 5, 4\}$
$\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$



$\mathcal{I}_1 = \{6, 1, 4, 2, 5, 3\}$
$\mathcal{I}_2 = \{1, 2, 4, 6, 3, 5\}$



Ingredient 2: restricted CPD algorithm

- ALS variant

$$\mathbf{A}_{k+1}^{(n)} = (1-\alpha)\mathbf{A}_{k}^{(n)} + \alpha \mathbf{T}_{(n)} \bar{\mathbf{V}}^{(n)} \left(\bar{\mathbf{W}}^{(n)}\right)^{-1}$$

Enforce restriction by $\alpha = \Delta_k$.

Ingredient 2: restricted CPD algorithm

- ALS variant

$$\mathbf{A}_{k+1}^{(n)} = (1-\alpha)\mathbf{A}_{k}^{(n)} + \alpha \mathbf{T}_{(n)} \bar{\mathbf{V}}^{(n)} \left(\bar{\mathbf{W}}^{(n)}\right)^{-1}$$

Enforce restriction by $\alpha = \Delta_k$.

- NLS variant

$$\min_{\mathbf{p}_k} \frac{1}{2} \left|\left|\text{vec}\left(\mathcal{F}(\mathbf{x}_k)\right) - \mathbf{J}_k \mathbf{p}_k\right|\right|^2 \text{ s.t. } ||\mathbf{p}_k|| \leq \Delta_k$$

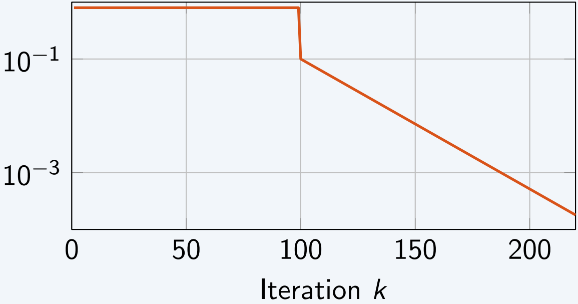
in which

$$\mathcal{F} = \mathcal{T} - \left[\!\left[\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \right]\!\right]$$

Ingredient 3: restriction

Use restriction of form

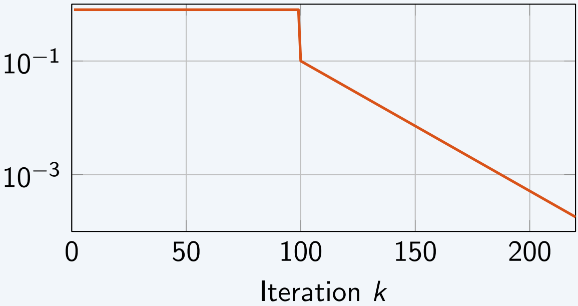
$$\Delta_k = \begin{cases} \Delta_0 & \text{if } k < K_{\text{search}} \\ \hat{\Delta}_0 \cdot \alpha^{(k-K_{\text{search}})/Q} & \text{if } k \geq K_{\text{search}} \end{cases}$$



Ingredient 3: restriction

Use restriction of form

$$\Delta_k = \begin{cases} \Delta_0 & \text{if } k < K_{\text{search}} \\ \hat{\Delta}_0 \cdot \alpha^{(k-K_{\text{search}})/Q} & \text{if } k \geq K_{\text{search}} \end{cases}$$

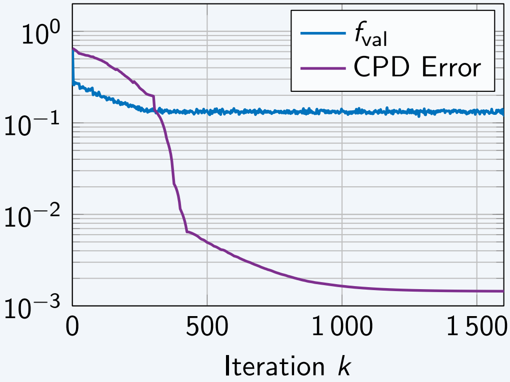


Example (Selecting Q)

For a $100 \times 100 \times 100$ tensor and block size $25 \times 25 \times 25$, $Q = 4$

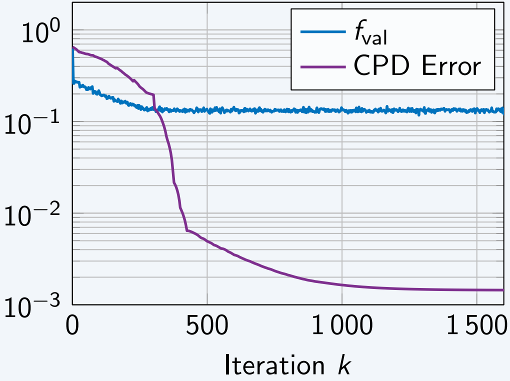
Ingredient 4: A stopping criterion

- Function evaluation $f_{\text{val}} = 0.5 \left|\left|\mathcal{T} - [\![\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}]\!]\right|\right|^2$



Ingredient 4: A stopping criterion

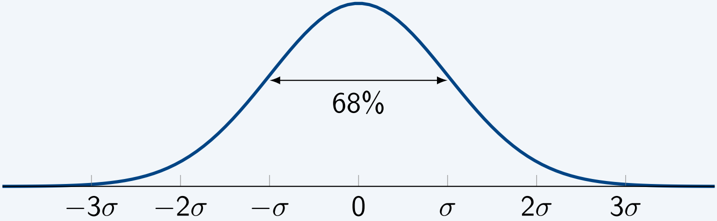
- Function evaluation $f_{\text{val}} = 0.5 \left|\left| \mathcal{T} - [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!] \right|\right|^2$



- Step size

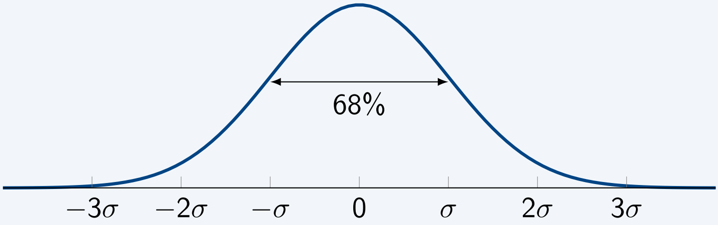
Intermezzo: Cramér–Rao bound

- Uncertainty of an estimate



Intermezzo: Cramér–Rao bound

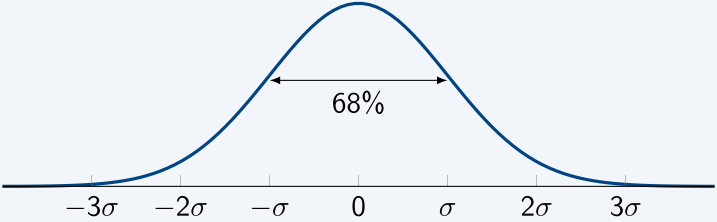
- Uncertainty of an estimate



- CRB $\leq \sigma^2$

Intermezzo: Cramér–Rao bound

- Uncertainty of an estimate



- CRB $\leq \sigma^2$
- $\mathbf{C} = \tau^2(\mathbf{J}^{\mathsf{H}}\mathbf{J})^{-1}$

Ingredient 4: Cramér–Rao bound based stopping criterion

- Experimental bound
  - Use estimates $\mathbf{A}_k^{(n)}$
  - Use f_{val} to estimate noise τ

Ingredient 4: Cramér–Rao bound based stopping criterion

- Experimental bound
  - Use estimates $\mathbf{A}_k^{(n)}$
  - Use f_{val} to estimate noise τ
- Stopping criterion:

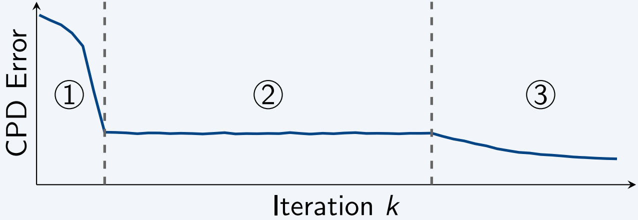
$$D_{\mathrm{CRB}} = \frac{1}{R\sum_n I_n} \sum_{n=1}^{N} \sum_{i=1}^{I_n} \sum_{r=1}^{R} \frac{\left| \mathbf{A}_k^{(n)}(i,r) - \mathbf{A}_{k-K_{\mathrm{CRB}}}^{(n)}(i,r) \right|}{\sqrt{\mathbf{C}^{(n)}(i,r)}}$$

Ingredient 4: Cramér–Rao bound based stopping criterion

- Experimental bound
  - Use estimates $\mathbf{A}_k^{(n)}$
  - Use f_{val} to estimate noise τ
- Stopping criterion:

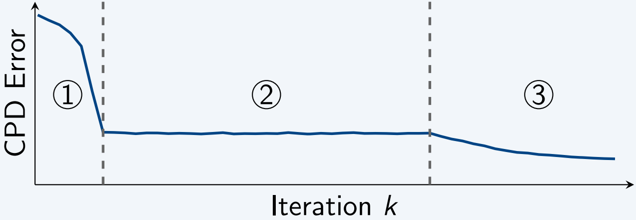
$$D_{\text{CRB}} = \frac{1}{R \sum_n I_n} \sum_{n=1}^{N} \sum_{i=1}^{I_n} \sum_{r=1}^{R} \frac{\left| \mathbf{A}_k^{(n)}(i,r) - \mathbf{A}_{k-K_{\text{CRB}}}^{(n)}(i,r) \right|}{\sqrt{\mathbf{C}^{(n)}(i,r)}}$$
$$\leq \gamma$$

Unrestricted phase vs restricted phase

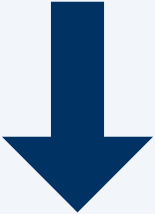


- Unrestricted phase (1 + 2): converge to a neighborhood of an optimum
- Restricted phase (3): pull iterates towards optimum

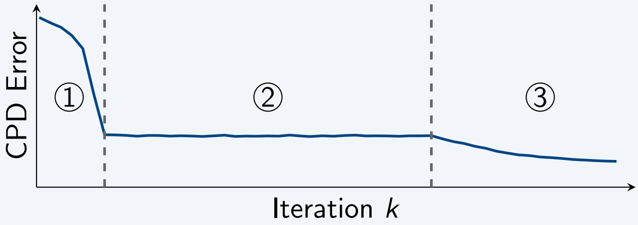
Unrestricted phase vs restricted phase



- Unrestricted phase (1 + 2): converge to a neighborhood of an optimum
- Restricted phase (3): pull iterates towards optimum



Unrestricted phase vs restricted phase



- Unrestricted phase $(1+2)$: converge to a neighborhood of an optimum
- Restricted phase (3): pull iterates towards optimum

Assumptions

- CPD of rank R exists
- SNR is high enough
- Most block dimensions $> R$

Overview

Experiment overview

- Experiments
  - Comparison ALS vs NLS (see paper)
  - Influence of block size
  - Influence of step size (see paper)

Experiment overview

- Experiments
  - Comparison ALS vs NLS (see paper)
  - Influence of block size
  - Influence of step size (see paper)
- Performance
  - 50 Monte Carlo experiments
  - CPD error

$$\max_n \left|\left|\mathbf{A}_0^{(n)} - \mathbf{A}_{\text{res}}^{(n)}\right|\right| / \left|\left|\mathbf{A}_0^{(n)}\right|\right|$$

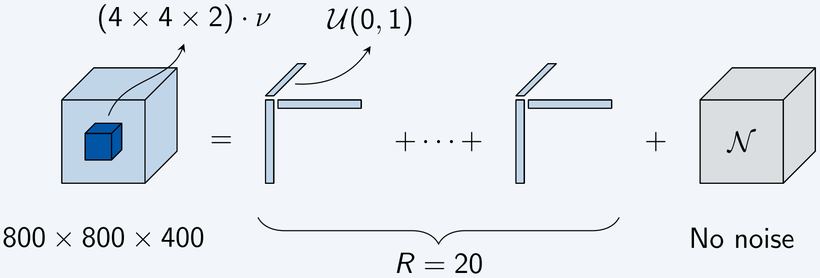
Experiment overview

- Experiments
  - Comparison ALS vs NLS (see paper)
  - Influence of block size
  - Influence of step size (see paper)
- Performance
  - 50 Monte Carlo experiments
  - CPD error

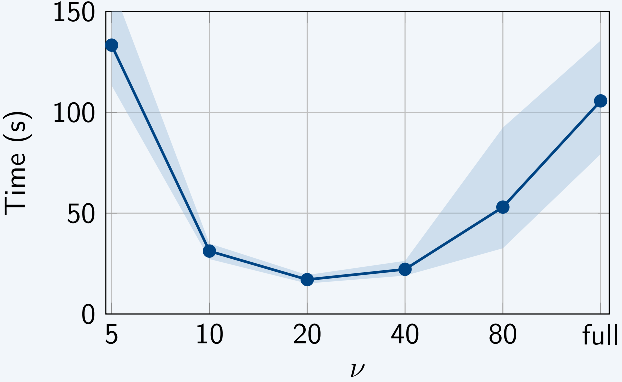
$$\max_n \left|\left|\mathbf{A}_0^{(n)} - \mathbf{A}_{\text{res}}^{(n)}\right|\right| / \left|\left|\mathbf{A}_0^{(n)}\right|\right|$$

- `cpd_rbs` in Tensorlab 3.0 [Vervliet et al. 2016c]

Influence of block size: setup

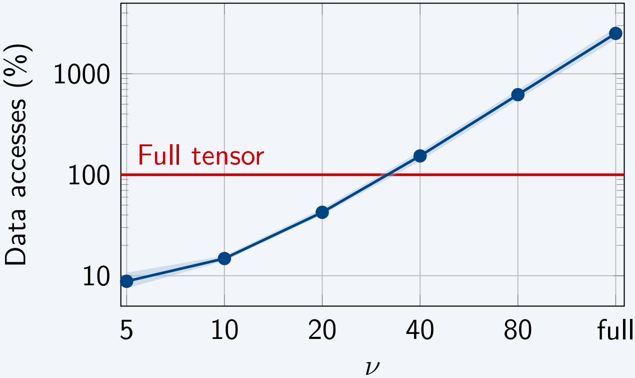


Influence of block size on computation time



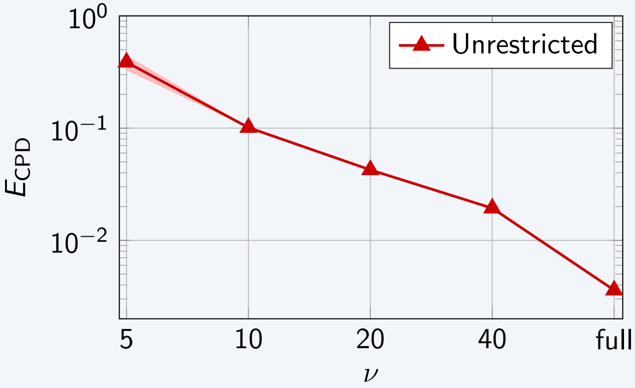
800 × 800 × 400 (4 × 4 × 2) · ν $R = 20$, $\mathcal{U}(0, 1)$ No noise

Influence of block size on data accesses



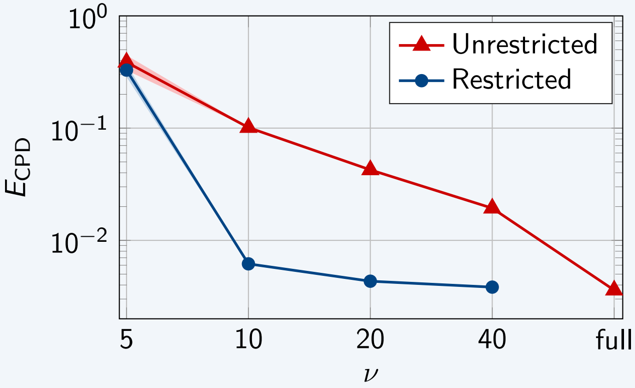
$800 \times 800 \times 400$ $(4 \times 4 \times 2) \cdot \nu$ $R = 20$, $\mathcal{U}(0, 1)$ No noise

Influence of block size on accuracy



$800 \times 800 \times 400$ $(4 \times 4 \times 2) \cdot \nu$ $R = 20$, $\mathcal{U}(0, 1)$ 20 dB

Influence of block size on accuracy



$800 \times 800 \times 400$ $(4 \times 4 \times 2) \cdot \nu$ $R = 20$, $\mathcal{U}(0,1)$ 20 dB

Overview

Decomposing large-scale tensors

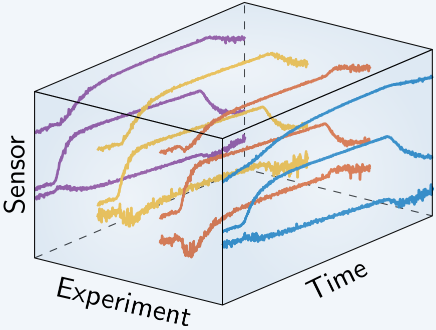
Randomized block sampling

Experimental results

Chemo-sensing application

Classify hazardous gasses

Does the sample contain CO, acetaldehyde or ammonia?

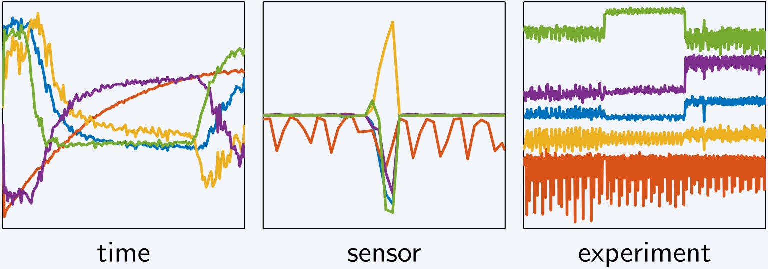


Strategy: classify using coefficients of spatiotemporal patterns.

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

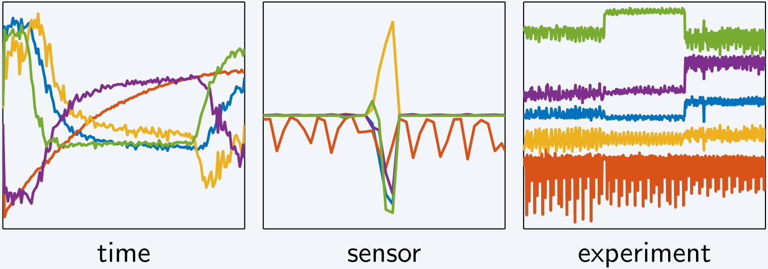
Classify hazardous gasses: results

- Resulting factor matrices



Classify hazardous gasses: results

- Resulting factor matrices



- Performance after clustering

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

Conclusion

- The randomized block sampling CPD algorithm enables the decomposition of larger tensors, using fewer data points and less memory
- Block size controls accuracy, data accesses and time
- Step size restriction improves accuracy
- Cramér–Rao bound based stopping criterion combines noise and step information

More details:

N. Vervliet and L. De Lathauwer [2016]. "A Randomized Block Sampling Approach to Canonical Polyadic Decomposition of Large-Scale Tensors". In: *IEEE Journal of Selected Topics in Signal Processing* 10.2, pp. 284–295

A randomized block sampling approach to the canonical polyadic decomposition of large-scale tensors

Nico Vervliet

Joint work with Lieven De Lathauwer

SIAM AN17, July 13, 2017

KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

fwo

erc
European Research Council

References I

- Acar, E., D.M. Dunlavy, and T.G. Kolda (2011a). "A scalable optimization approach for fitting canonical tensor decompositions". In: *Journal of Chemometrics* 25.2, pp. 67–86.
- Acar, E. et al. (2011b). "Scalable tensor factorizations for incomplete data". In: *Chemometrics and Intelligent Laboratory Systems* 106.1, pp. 41–56.
- Bader, B.W. and T.G. Kolda (2007). "Efficient MATLAB computations with sparse and factored tensors". In: *SIAM J. Sci. Comput.* 30.1, pp. 205–231.
- Kang, U. et al. (2012). "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, pp. 316–324.

References II

- Liavas, A. and N. Sidiropoulos (2015). "Parallel Algorithms for Constrained Tensor Factorization via the Alternating Direction Method of Multipliers". In: *IEEE Trans. Signal Process.* PP.99, pp. 1–1.
- Oseledets, I.V. and E.E. Tyrtyshnikov (2010). "TT-cross approximation for multidimensional arrays". In: *Linear Algebra and its Applications* 432.1, pp. 70–88.
- Papalexakis, E., C. Faloutsos, and N. Sidiropoulos (2012). "ParCube: Sparse Parallelizable Tensor Decompositions". English. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by PeterA. Flach, Tijl De Bie, and Nello Cristianini. Vol. 7523. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 521–536.

References III

Phan, A.-H. and A. Cichocki (2011). “PARAFAC algorithms for large-scale problems”. In: *Neurocomputing* 74.11, pp. 1970–1984.

Phan, A.-H., P. Tichavský, and A. Cichocki (2013). “Low Complexity Damped Gauss–Newton Algorithms for CANDECOMP/PARAFAC”. In: *SIAM J. Appl. Math.* 34.1, pp. 126–147.

Sidiropoulos, N., E. Papalexakis, and C. Faloutsos (2014). “Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition”. In: *IEEE Signal Process. Mag.* 31.5, pp. 57–70.

- Sorber, L., M. Van Barel, and L. De Lathauwer (2013). "Optimization-Based Algorithms for Tensor Decompositions: Canonical Polyadic Decomposition, Decomposition in Rank-$(L_r, L_r, 1)$ Terms, and a New Generalization". In: 23.2, pp. 695–720.
- Vervliet, N. and L. De Lathauwer (2016). "A Randomized Block Sampling Approach to Canonical Polyadic Decomposition of Large-Scale Tensors". In: *IEEE Journal of Selected Topics in Signal Processing* 10.2, pp. 284–295.
- Vervliet, N., O. Debals, and L. De Lathauwer (2016a). "Canonical polyadic decomposition of incomplete tensors with linearly constrained factors". Technical Report 16–172, ESAT-STADIUS, KU Leuven, Belgium.

– (2016b). "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*.

Vervliet, N. et al. (2014). "Breaking the Curse of Dimensionality Using Decompositions of Incomplete Tensors: Tensor-based scientific computing in big data analysis". In: *IEEE Signal Process. Mag.* 31.5, pp. 71–79.

Vervliet, N. et al. (2016c). *Tensorlab 3.0*. Available online at http://www.tensorlab.net.