Matrix and Tensor Functions in Conflict with Approximation

Lars Grasedyck
(Christian Löbbert, Lukas Juschka)

Institut für Geometrie und Praktische Mathematik
RWTH Aachen University

Outline

Low Rank Matrices and Tensors

The Hierarchical Tucker Format

Determine Largest Element

Interesting Problems

Low Rank Matrices and Tensors

Our objects of interest are multivariate functions

$$f(x_1, \dots, x_d) \in \mathbb{R}, \qquad x \in I = I_1 \times \cdots \times I_d$$

with typically **finite** univariate index sets I_μ

Low Rank Matrices and Tensors

Our objects of interest are multivariate functions

$$f(x_1, \ldots, x_d) \in \mathbb{R}, \qquad x \in I = I_1 \times \cdots \times I_d$$

with typically **finite** univariate index sets I_μ

We want to represent or approximate the function in low rank form

$$f(x_1, \ldots, x_d) \approx \sum_j f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

with **univariate** $f_{j,\mu}$

Low Rank Matrices and Tensors

Our objects of interest are multivariate functions

$$f(x_1, \ldots, x_d) \in \mathbb{R}, \qquad x \in I = I_1 \times \cdots \times I_d$$

with typically **finite** univariate index sets I_μ

We want to represent or approximate the function in low rank form

$$f(x_1, \ldots, x_d) \approx \sum_j f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

with **univariate** $f_{j,\mu}$

- Data-sparse representation of each $f_{j,\mu}$ easy
- Curse of dimensionality lifted **provided few summands**
- Need to find/determine the $f_{j,\mu}$

Low Rank Matrices and Tensors

$$f(x_1, \ldots, x_d) \approx \sum_j f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

Is this format useful for computations?

Low Rank Matrices and Tensors

$$f(x_1, \ldots, x_d) \approx \sum_j f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

Is this format useful for computations?

Hilbert-Schmidt (SVD) in $d = 2$

$$f(x_1, x_2) = \sum_{j=1}^{r} f_{j,1}(x_1) f_{j,2}(x_2)$$

s.t. $f_{\cdot,1}$ and $f_{\cdot,2}$ orthogonal / normalized

Low Rank Matrices and Tensors

$$f(x_1, \ldots, x_d) \approx \sum_j f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

Is this format useful for computations?

Hilbert-Schmidt (SVD) in $d = 2$

$$f(x_1, x_2) = \sum_{j=1}^{r} f_{j,1}(x_1) f_{j,2}(x_2)$$

s.t. $f_{\cdot,1}$ and $f_{\cdot,2}$ orthogonal / normalized

Problem: This framework exists **only** in $d < 3$

Low Rank Matrices and Tensors

$$f(x_1,\ldots,x_d) \approx \sum_j f_{j,1}(x_1)\cdots f_{j,d}(x_d)$$

Is this format useful for computations?

Hilbert-Schmidt (SVD) in $d = 2$

$$f(x_1,x_2) = \sum_{j=1}^{r} f_{j,1}(x_1) f_{j,2}(x_2)$$

s.t. $f_{\cdot,1}$ and $f_{\cdot,2}$ orthogonal / normalized

Problem: This framework exists **only** in $d < 3$

Our goal is to extend this to high dimension $d \gg 1000$

Example: d-dimensional Laplacian

We solve [Computing 2004, G.]

$$-\Delta u(x) = 1, \qquad x \in \Omega = [0,1]^d, \qquad u|_{\partial\Omega} = 0$$

on a grid with 1024^d grid points and rel. pointwise accuracy $\approx 10^{-5}$

Example: d-dimensional Laplacian

We solve [Computing 2004, G.]

$$-\Delta u(x) = 1, \qquad x \in \Omega = [0,1]^d, \qquad u|_{\partial\Omega} = 0$$

on a grid with 1024^d grid points and rel. pointwise accuracy $\approx 10^{-5}$

d	128	256	512	1024	2048	4096
Seconds	32	31	35	48	82	187

Example: d-dimensional Laplacian

We solve [Computing 2004, G.]

$$-\Delta u(x) = 1, \qquad x \in \Omega = [0,1]^d, \qquad u|_{\partial\Omega} = 0$$

on a grid with 1024^d grid points and rel. pointwise accuracy $\approx 10^{-5}$

d	128	256	512	1024	2048	4096
Seconds	32	31	35	48	82	187

Why do we choose the grid/basis fixed?

Example: d-dimensional Laplacian

We solve [Computing 2004, G.]

$$-\Delta u(x) = 1, \qquad x \in \Omega = [0,1]^d, \qquad u|_{\partial\Omega} = 0$$

on a grid with 1024^d grid points and rel. pointwise accuracy $\approx 10^{-5}$

d	128	256	512	1024	2048	4096
Seconds	32	31	35	48	82	187

Why do we choose the grid/basis fixed?

ease of presentation, cf. [FOCM 16,2016, Dahmen,DeVore,G.,Süli]
Results based on exponential sums $\frac{1}{x} \approx \sum_{j=1}^{r} w_j \exp(t_j x)$

Example: d-dimensional Laplacian

$$-\Delta u(x) = 1, \qquad x \in \Omega = [0,1]^d, \qquad u|_{\partial\Omega} = 0$$

The operator discretized leads to a (sparse) matrix A and we solve

$$Ax = b$$

We apply the matrix function

$$F(A) = \sum_{j=1}^{r} w_j \exp(t_j A) \approx A^{-1}$$

and obtain — for structured b — a structured solution

$$x \approx \tilde{x} = \sum_{j=1}^{r} w_j \exp(t_j A) b$$

Institut für Geometrie und Praktische Mathematik

Low Rank Matrices and Tensors

Remarks:

1. The operator (and thus A) is structured

Low Rank Matrices and Tensors

Remarks:

2. The right-hand side is assumed to be structured

Low Rank Matrices and Tensors

Remarks:

3. We require an explicit a priori exponential sum approximation

$$\frac{1}{x} \approx \sum_{j=1}^{r} w_j \exp(t_j x), \qquad x \in (1, \infty)$$

Low Rank Matrices and Tensors

Remarks:

4. This trick is so good, it is used in all sciences

Low Rank Matrices and Tensors

Remarks:

1. The operator (and thus A) is structured
2. The right-hand side is assumed to be structured
3. We require an explicit a priori exponential sum approximation

$$\frac{1}{x} \approx \sum_{j=1}^{r} w_j \exp(t_j x), \qquad x \in (1, \infty)$$

4. This trick is so good, it is used in all sciences
5. There exists essentially no generalisation (so far?)

Low Rank Matrices and Tensors

Remarks:

1. The operator (and thus A) is structured
2. The right-hand side is assumed to be structured
3. We require an explicit a priori exponential sum approximation

$$\frac{1}{x} \approx \sum_{j=1}^{r} w_j \exp(t_j x), \qquad x \in (1, \infty)$$

4. This trick is so good, it is used in all sciences
5. There exists essentially no generalisation (so far?)

$$\boxed{\text{We need to change the low rank format}}$$

The Hierarchical Tucker Format

Low rank matrix representation $M = UV^T$

$$M(i_1, i_2) = \sum_{j=1}^{k} U(i_1, j) V(i_2, j)$$

Institut für Geometrie und Praktische Mathematik

The Hierarchical Tucker Format

Low rank matrix representation $M = UV^T$

$$M(i_1, i_2) = \sum_{j=1}^{k} U(i_1, j) V(i_2, j)$$

Low rank tensor representation (CP) format

$$M(i_1, \ldots, i_d) = \sum_{j=1}^{k} M_{j,1}(i_1) \cdots M_{j,d}(i_d)$$

The Hierarchical Tucker Format

Low rank matrix representation $M = UV^T$

$$M(i_1, i_2) = \sum_{j=1}^{k} U(i_1, j) V(i_2, j)$$

Low rank tensor representation (CP) format

$$M(i_1, \ldots, i_d) = \sum_{j=1}^{k} M_{j,1}(i_1) \cdots M_{j,d}(i_d)$$

Matrices of rank $\leq k$ closed

The Hierarchical Tucker Format

Low rank matrix representation $M = UV^T$

$$M(i_1, i_2) = \sum_{j=1}^{k} U(i_1, j) V(i_2, j)$$

Low rank tensor representation (CP) format

$$M(i_1, \ldots, i_d) = \sum_{j=1}^{k} M_{j,1}(i_1) \cdots M_{j,d}(i_d)$$

Matrices of rank $\leq k$ closed

Tensors of rank $\leq k$ not closed (and complicated)

The Hierarchical Tucker Format

Low rank matrix representation $M = UV^T$

$$M(i_1, i_2) = \sum_{j=1}^{k} U(i_1, j)V(i_2, j)$$

Low rank tensor representation (CP) format

$$M(i_1, \ldots, i_d) = \sum_{j=1}^{k} M_{j,1}(i_1) \cdots M_{j,d}(i_d)$$

- Matrices of rank $\leq k$ closed
- Tensors of rank $\leq k$ not closed (and complicated)
- CP decomposition used e.g. in blind source separation

The Hierarchical Tucker Format

Idea (Hackbusch): Use a hierarchy of low rank matrices:

$$M(\boxed{i_1, i_2}, \boxed{i_3, i_4}) = \sum_{j=1}^{k} U^{(1,2)}(\boxed{i_1, i_2}, j)\, U^{(3,4)}(\boxed{i_3, i_4}, j)$$

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

The Hierarchical Tucker Format

Idea (Hackbusch): Use a hierarchy of low rank matrices:

$$M(\boxed{i_1, i_2}, \boxed{i_3, i_4}) = \sum_{j=1}^{k} U^{(1,2)}(\boxed{i_1, i_2}, j) U^{(3,4)}(\boxed{i_3, i_4}, j)$$

$$U^{(1,2)}(i_1, i_2, j) = \sum_{\ell=1}^{k} \sum_{m=1}^{k} B^{(1,2)}{}_{j,\ell,m} U^{(1)}(i_1, \ell) U^{(2)}(i_2, m)$$

$$U^{(3,4)}(i_3, i_4, j) = \sum_{\ell=1}^{k} \sum_{m=1}^{k} B^{(3,4)}{}_{j,\ell,m} U^{(3)}(i_3, \ell) U^{(4)}(i_4, m)$$

The Hierarchical Tucker Format

Idea (Hackbusch): Use a hierarchy of low rank matrices:

$$M(\boxed{i_1, i_2}, \boxed{i_3, i_4}) = \sum_{j=1}^{k} U^{(1,2)}(\boxed{i_1, i_2}, j) U^{(3,4)}(\boxed{i_3, i_4}, j)$$

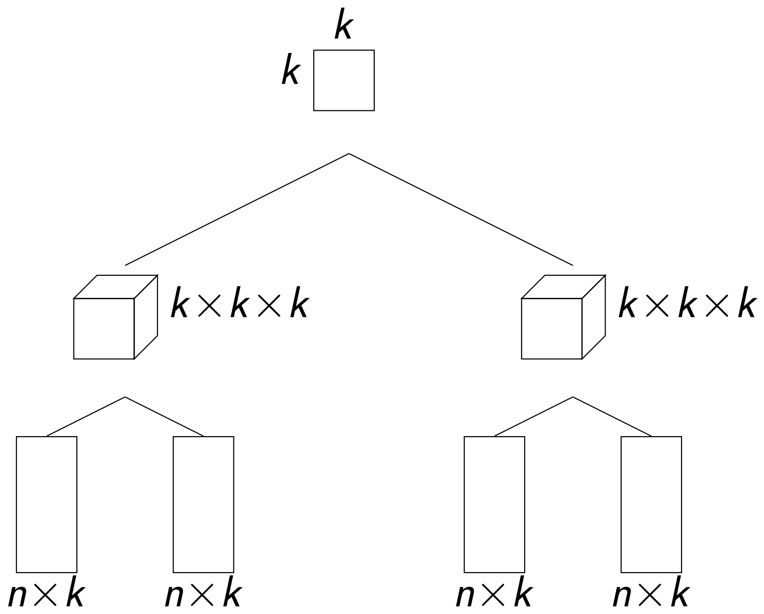
$$U^{(1,2)}(i_1, i_2, j) = \sum_{\ell=1}^{k} \sum_{m=1}^{k} B^{(1,2)}{}_{j,\ell,m} U^{(1)}(i_1, \ell) U^{(2)}(i_2, m)$$

$$U^{(3,4)}(i_3, i_4, j) = \sum_{\ell=1}^{k} \sum_{m=1}^{k} B^{(3,4)}{}_{j,\ell,m} U^{(3)}(i_3, \ell) U^{(4)}(i_4, m)$$

$\rightarrow$ (HT) format

HT Tensors of rank $\leq k$ closed (and simple)

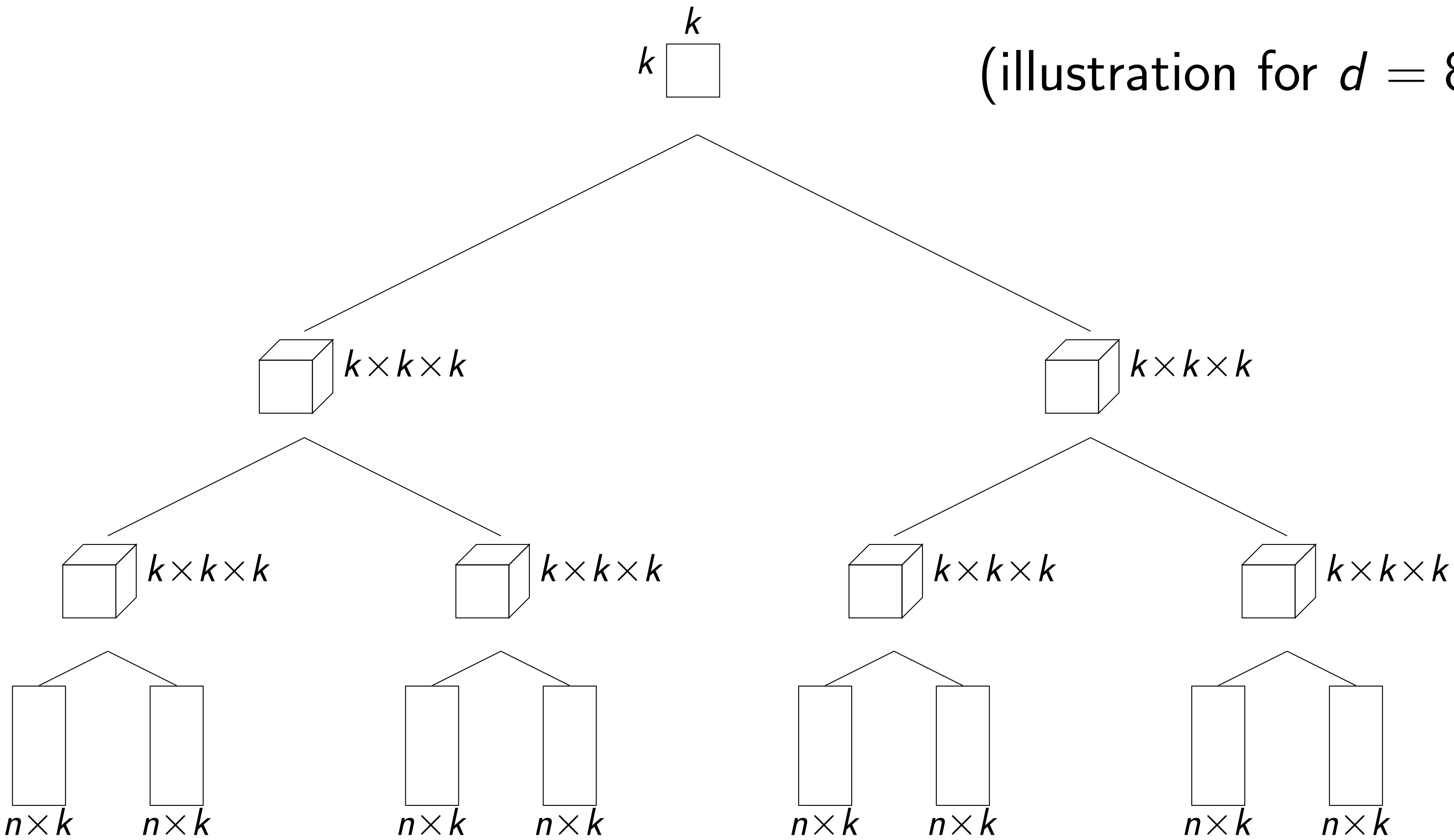
The Hierarchical Tucker Format



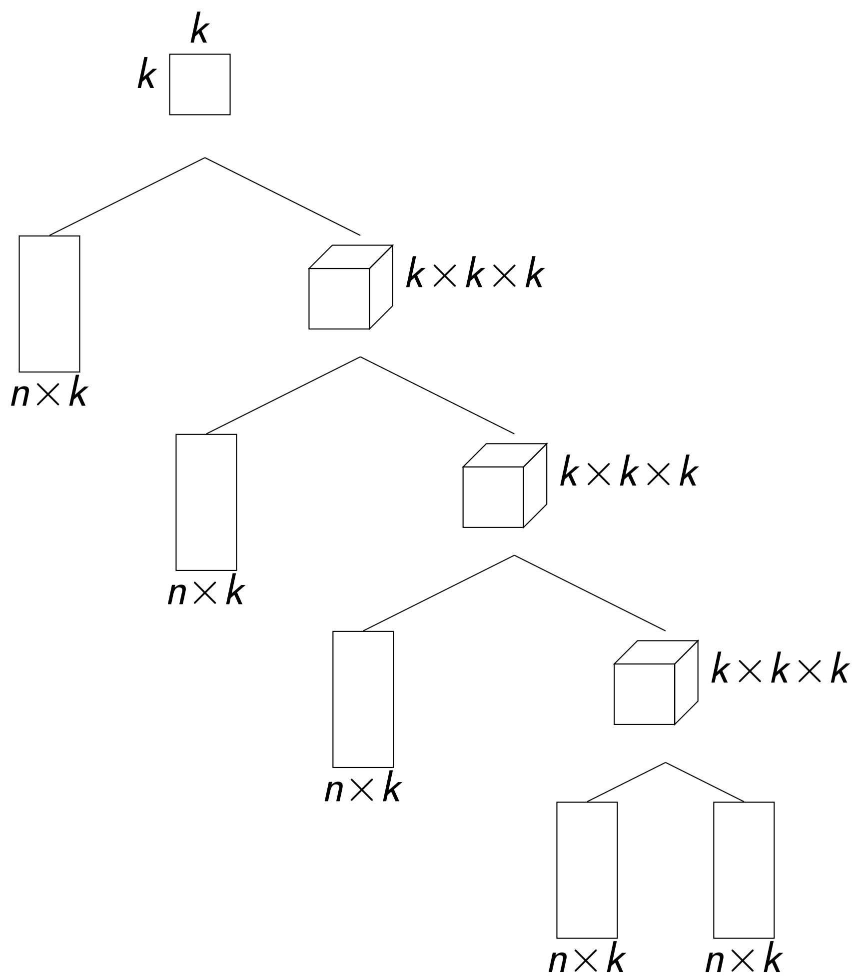
(illustration for $d = 4$)

The Hierarchical Tucker Format

(illustration for $d = 8$)



The Hierarchical Tucker Format



(illustration for $d = 5$)

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n \times \cdots \times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n \times \cdots \times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n\times\cdots\times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$
full tensor:

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n\times\cdots\times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$
full tensor: $100,000,000,000,000,000,000$ entries

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n\times\cdots\times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$
full tensor: $100,000,000,000,000,000,000$ entries
HT tensor:

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n\times\cdots\times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2)\cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$

full tensor:	$100,000,000,000,000,000,000$	entries
HT tensor:	$3,275$	parameters

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n \times \cdots \times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20$, $n = 10$, $k = 5$

full tensor:	$100,000,000,000,000,000,000$	entries
HT tensor:	$3,275$	parameters
CP tensor:		

The Hierarchical Tucker Format

A d-dimensional tensor $M \in \mathbb{R}^{n \times \cdots \times n}$ (i.e. n^d entries) of **low rank** k allows for a representation in the Hierarchical Tucker Format with only

$$d \cdot n \cdot k + (d-2) \cdot k^3 + k^2 = \mathcal{O}(d) \text{ entries.}$$

Example: $d = 20,\ n = 10,\ k = 5$

full tensor:	$100,000,000,000,000,000,000$	entries
HT tensor:	$3,275$	parameters
CP tensor:	$1,000$	parameters

The Hierarchical Tucker Format

Matricization and low rank representation for $t = \{2, 3, 5\}$:

$$M(\boxed{i_2, i_3, i_5}, \boxed{i_1, i_4, i_6, i_7}) = \sum_{j=1}^{k} U(\boxed{i_2, i_3, i_5}, j) V(\boxed{i_1, i_4, i_6, i_7}, j)$$

The Hierarchical Tucker Format

Matricization and low rank representation for $t = \{2, 3, 5\}$:

$$M(\boxed{i_2, i_3, i_5}, \boxed{i_1, i_4, i_6, i_7}) = \sum_{j=1}^{k} U(\boxed{i_2, i_3, i_5}, j) V(\boxed{i_1, i_4, i_6, i_7}, j)$$

For each t this defines a set $R(t, k) \subset \mathbb{R}^{n^d}$.

The Hierarchical Tucker Format

Matricization and low rank representation for $t = \{2, 3, 5\}$:

$$M(\boxed{i_2, i_3, i_5}, \boxed{i_1, i_4, i_6, i_7}) = \sum_{j=1}^{k} U(\boxed{i_2, i_3, i_5}, j) V(\boxed{i_1, i_4, i_6, i_7}, j)$$

For each t this defines a set $R(t, k) \subset \mathbb{R}^{n^d}$.

$$HT(k) := \bigcap_{t \in T} R(t, k) \qquad \boxed{\text{tree } T}$$

The Hierarchical Tucker Format

Matricization and low rank representation for $t = \{2, 3, 5\}$:

$$M(\boxed{i_2, i_3, i_5}, \boxed{i_1, i_4, i_6, i_7}) = \sum_{j=1}^{k} U(\boxed{i_2, i_3, i_5}, j) V(\boxed{i_1, i_4, i_6, i_7}, j)$$

For each t this defines a set $R(t, k) \subset \mathbb{R}^{n^d}$.

$$HT(k) := \bigcap_{t \in T} R(t, k) \qquad \boxed{\text{tree } T}$$

Two main research directions:

1. Design algorithms to work with and exploit data-sparse tensor representations ($\gg$ 1000 articles)
2. Prove approximability / convergence (few articles)

Truncation of Hierarchical Tensors / Projection onto $HT(k)$

The computation of $u + v$ doubles the rank: $k \rightarrow 2k$.

Truncation of Hierarchical Tensors / Projection onto $HT(k)$

The computation of $u + v$ doubles the rank: $k \to 2k$.
The computation of $\mathbf{A}u$ or $u \circ v$ even causes $k \to k^2$.

Truncation of Hierarchical Tensors / Projection onto $HT(k)$

The computation of $u + v$ doubles the rank: $k \to 2k$.
The computation of $\mathbf{A}u$ or $u \circ v$ even causes $k \to k^2$.

A tensor u can either be truncated
– to a tensor $\widetilde{u}$ of prescribed rank k or

Truncation of Hierarchical Tensors / Projection onto $HT(k)$

The computation of $u + v$ doubles the rank: $k \to 2k$.
The computation of $\mathbf{A}u$ or $u \circ v$ even causes $k \to k^2$.

A tensor u can either be truncated

- to a tensor $\widetilde{u}$ of prescribed rank k or
- to a tensor $\widetilde{u}$, that fulfills $\|u - \widetilde{u}\| < \varepsilon$ for a prescribed error ε (the rank of $\widetilde{u}$ gets chosen adaptively).

Truncation of Hierarchical Tensors / Projection onto $HT(k)$

The computation of $u + v$ doubles the rank: $k \to 2k$.
The computation of $\mathbf{A}u$ or $u \circ v$ even causes $k \to k^2$.

A tensor u can either be truncated
- to a tensor $\widetilde{u}$ of prescribed rank k or
- to a tensor $\widetilde{u}$, that fulfills $\|u - \widetilde{u}\| < \varepsilon$ for a prescribed error ε (the rank of $\widetilde{u}$ gets chosen adaptively).

(Truncation should be applied after a certain amount of summations or multiplications to keep the amount of data $dnk + (d-2)k^3 + k^2$ below some upper limit)

Truncation of Hierarchical Tensors

Truncation of Hierarchical Tensors

Best approximation of $u \in HT(T, k)$ not computable

Truncation of Hierarchical Tensors

Best approximation of $u \in HT(T, k)$ not computable

Independent (best) truncation in each $R(T, k)$ $(HT(T, k) = \cap_{t \in T} R(t, k))$ yields

$$\|u - \widetilde{u}\| \leq \sqrt{2d-3} \inf_{v \in HT(T,k)} \|u - v\|$$

Truncation of Hierarchical Tensors

Best approximation of $u \in HT(T, k)$ not computable

Independent (best) truncation in each $R(T, k)$ $(HT(T, k) = \cap_{t \in T} R(t, k))$ yields

$$\|u - \widetilde{u}\| \leq \sqrt{2d - 3} \inf_{v \in HT(T,k)} \|u - v\|$$

Speed of truncation (single processor CPU, 9 years ago):

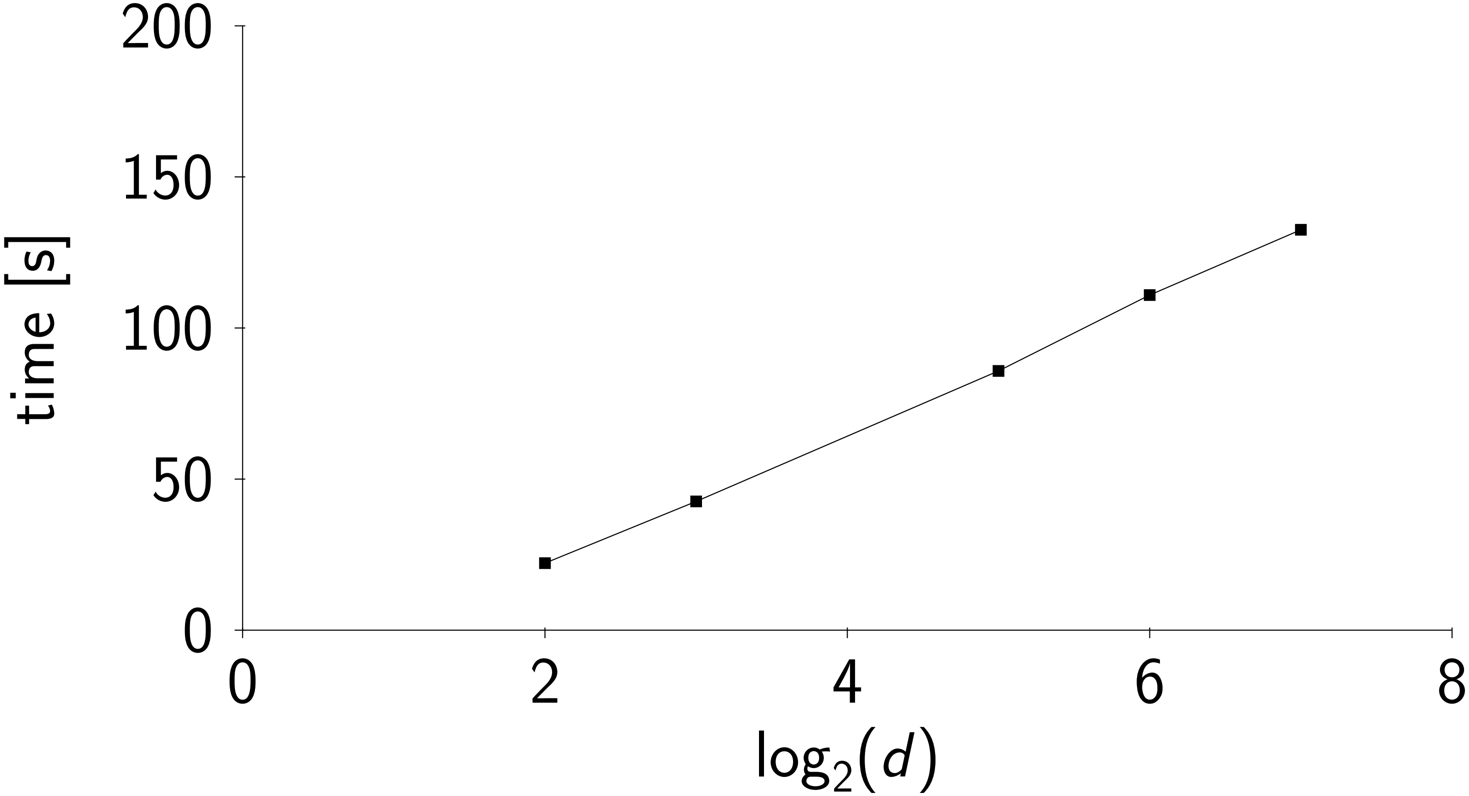
$n = 100$, $d = 1000$, input rank $k = 25$

Memory consumption: 138 MB

Time consumption: 19 seconds

Parallel Truncation [Num.Lin.Alg.Appl., G.,Löbbert, acc.]

time $\sim \log_2(d)$ (full tensor: 10^{512} entries)



The Hierarchical Tucker Format

Postprocessing with low rank tensors can be easy:

$$QoI = \int f(x) d\pi(x)$$

with product measure π and

$$f(x_1, \ldots, x_d) = \sum_{j=1}^{k} f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

gives

$$QoI = \sum_{j=1}^{k} \prod_{\mu=1}^{d} \int f_{j,\mu}(x_\mu) d\pi_\mu(x_\mu)$$

The Hierarchical Tucker Format

Postprocessing with low rank tensors can be easy:

$$QoI = \int f(x) d\pi(x)$$

with product measure π and

$$f(x_1, \dots, x_d) = \sum_{j=1}^{k} f_{j,1}(x_1) \cdots f_{j,d}(x_d)$$

gives

$$QoI = \sum_{j=1}^{k} \prod_{\mu=1}^{d} \int f_{j,\mu}(x_\mu) d\pi_\mu(x_\mu)$$

...or complicated:

$$\max_{i_1,\dots,i_d} f(i_1, \dots, i_d) = ?$$

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

Determine Largest Element

$$\max_{i_1,\ldots,i_d} f(i_1,\ldots,i_d) = ?$$

1. Conditioning of the problem
2. Numerical heuristics

Determine Largest Element

1. Conditioning

$$M := \max_{i_1,\dots,i_d} |f(i_1,\dots,i_d)| =?$$

Any elementwise perturbation E of size ε yields

$$\max_{i_1,\dots,i_d} |f(i_1,\dots,i_d) + E(i_1,\dots,i_d)| \in [M-\varepsilon, M+\varepsilon]$$

1. Conditioning

$$M := \max_{i_1,\dots,i_d} |f(i_1,\dots,i_d)| =?$$

Any elementwise perturbation E of size ε yields

$$\max_{i_1,\dots,i_d} |f(i_1,\dots,i_d) + E(i_1,\dots,i_d)| \in [M-\varepsilon, M+\varepsilon]$$

What about

$$\max_{i_1,\dots,i_d} |f(i_1,\dots,i_d) + E(i_1,\dots,i_d)|$$

for $\|M\| \le \varepsilon \|f\|$?

Determine Largest Element

$$\max_{i_1,\ldots,i_d} |f(i_1,\ldots,i_d) + E(i_1,\ldots,i_d)| \in ?, \qquad \|M\| \leq \varepsilon \|f\|$$

Example: $\varepsilon \sim 1/\sqrt{n}$

Determine Largest Element

$$\max_{i_1,\dots,i_d} |f(i_1,\dots,i_d) + E(i_1,\dots,i_d)| \in ?, \qquad \|M\| \le \varepsilon \|f\|$$

Example: $\varepsilon \sim 1/\sqrt{n}$

$$f = \left[\begin{array}{ccc|c} \varepsilon & \cdots & \varepsilon & 0 \\ \vdots & \ddots & \vdots & 0 \\ \varepsilon & \cdots & \varepsilon & 0 \\ \hline 0 & \cdots & 0 & 1 \end{array}\right] \in \mathbb{R}^{n+1 \times n+1}, \qquad \|f\|_F = \sqrt{n^2\varepsilon^2 + 1} \sim \sqrt{n}$$

$$M = \left[\begin{array}{ccc|c} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 0 & 0 \\ \hline 0 & \cdots & 0 & -1 \end{array}\right] \in \mathbb{R}^{n+1 \times n+1}, \qquad \|M\|_F = 1 \sim \varepsilon \|f\|_F$$

In higher dimension d ill-conditioned in $\|\cdot\|_F$

Determine Largest Element

$$\max_{i_1,\ldots,i_d} |f(i_1,\ldots,i_d) + E(i_1,\ldots,i_d)| \in ?, \qquad \|M\| \leq \varepsilon \|f\|$$

Example: $\varepsilon \sim n^{-d/4}$

Determine Largest Element

$$\max_{i_1,\dots,i_d} |f(i_1,\dots,i_d) + E(i_1,\dots,i_d)| \in ?, \qquad \|M\| \le \varepsilon \|f\|$$

Example: $\varepsilon \sim n^{-d/4}$

$$f = \varepsilon v \otimes \cdots \otimes v \in \mathbb{R}^{(n+1)^d}, \qquad \|f\|_F = \sqrt{n^d \varepsilon^2 + 1} \sim n^{d/4}$$

$$M = -e_{n+1} \otimes \cdots \otimes e_{n+1} \in \mathbb{R}^{(n+1)^d}, \qquad \|M\|_F = 1 \sim \varepsilon \|f\|_F$$

In higher dimension d ill-conditioned in $\|\cdot\|_F$

2. Numerical Heuristics

First approach:

for any rank 1 tensor $v = v_1 \otimes \cdots \otimes v_d$ we have

$$\|v\|_\infty = \prod_{\mu=1}^{d} \|v_\mu\|_\infty$$

2. Numerical Heuristics

First approach:

for any rank 1 tensor $v = v_1 \otimes \cdots \otimes v_d$ we have

$$\|v\|_\infty = \prod_{\mu=1}^{d} \|v_\mu\|_\infty$$

$$\mathrm{argmax}_i |v(i)| = \left(\mathrm{argmax}_{i_\mu} |v_\mu(i_\mu)|\right)_{\mu=1}^{d}$$

2. Numerical Heuristics

First approach:

for any rank 1 tensor $v = v_1 \otimes \cdots \otimes v_d$ we have

$$\|v\|_\infty = \prod_{\mu=1}^{d} \|v_\mu\|_\infty$$

$$\mathrm{argmax}_i |v(i)| = (\mathrm{argmax}_{i_\mu} |v_\mu(i_\mu)|)_{\mu=1}^{d}$$

truncate tensor M to rank 1 tensor $\tilde{M}$ and compute its max

2. Numerical Heuristics

First approach: truncate to rank 1
Example: $\varepsilon \sim n^{-d/2}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

2. Numerical Heuristics

First approach: truncate to rank 1
Example: $\varepsilon \sim n^{-d/2}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

Rank 1 best approximation in $\|\cdot\|_F$ is

$$\tilde{M} = \varepsilon v \otimes \cdots \otimes v$$

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

2. Numerical Heuristics

First approach: truncate to rank 1
Example: $\varepsilon \sim n^{-d/2}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

Rank 1 best approximation in $\|\cdot\|_F$ is

$$\tilde{M} = \varepsilon v \otimes \cdots \otimes v$$

Result: $\|M\|_\infty = 1$ estimated by $\|\tilde{M}\|_\infty = n^{-d/2}$,
gives completely wrong answer

2. Numerical Heuristics

Second approach: power iteration (Mike Espig)

compute Hadamard products

$$\hat{M} := M \odot M \odot M \cdots$$

2. Numerical Heuristics

Second approach: power iteration (Mike Espig)

compute Hadamard products

$$\hat{M} := M \odot M \odot M \cdots$$

truncate tensor $\hat{M}$ to rank 1 tensor $\tilde{M}$ and compute its argmax

2. Numerical Heuristics

Second approach: power iteration
Example: $\varepsilon \sim n^{-1}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

Determine Largest Element

2. Numerical Heuristics

Second approach: power iteration
Example: $\varepsilon \sim n^{-1}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

$$\hat{M} := M \odot M \odot M \cdots = \varepsilon^{\ell} v \otimes \cdots \otimes v$$

2. Numerical Heuristics

Second approach: power iteration
Example: $\varepsilon \sim n^{-1}$

$$M = \varepsilon v \otimes \cdots \otimes v + e_{n+1} \otimes \cdots \otimes e_{n+1}$$

$$\hat{M} := M \odot M \odot M \cdots = \varepsilon^{\ell} v \otimes \cdots \otimes v$$

Rank 1 best approximation in $\|\cdot\|_F$ is

$$\tilde{M} = \varepsilon^{\ell} v \otimes \cdots \otimes v$$

for all $\ell < d/2$

Determine Largest Element

2. Numerical Heuristics

Further approaches:

- subspace iterations (Lukas Juschka)
- local search
- many optimizations. . .

[cf. Higham and Relton 2015]

Determine Largest Element

2. Numerical Heuristics

Further approaches:

- subspace iterations (Lukas Juschka)
- local search
- many optimizations...

[cf. Higham and Relton 2015]

Lower bound: largest known value
Good upper bound: ? some p-norm
$\rightarrow$ branach and bound

(work in progress)

igpm Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

Interesting Problems

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T ($\rightarrow$ Ballani, G. SISC 36, 2014)

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T ($\rightarrow$ Ballani, G. SISC 36, 2014)
4. A priori assess $u \approx \tilde{u} \in HT(T, k)$

Interesting Problems

1. Stability of Tensor Completion
   (→ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T (→ Ballani, G. SISC 36, 2014)
4. A priori assess $u \approx \tilde{u} \in HT(T, k)$
5. Postprocess $\tilde{u}$ to find max, min, cuts, ...

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T ($\rightarrow$ Ballani, G. SISC 36, 2014)
4. A priori assess $u \approx \tilde{u} \in HT(T, k)$
5. Postprocess $\tilde{u}$ to find max, min, cuts, ...
6. Prove (global) convergence of alternating minimization

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T ($\rightarrow$ Ballani, G. SISC 36, 2014)
4. A priori assess $u \approx \tilde{u} \in HT(T, k)$
5. Postprocess $\tilde{u}$ to find max, min, cuts, ...
6. Prove (global) convergence of alternating minimization
7. From trees to networks

Institut für Geometrie und Praktische Mathematik
RWTH AACHEN UNIVERSITY

Interesting Problems

1. Stability of Tensor Completion
   ($\rightarrow$ Krämer, G. arXiv:1701.08045) and other operations
2. Geometry of $HT(T, k)$,
   e.g. feasible singular values, Krämer arXiv:1701.08437
3. Finding the tree T ($\rightarrow$ Ballani, G. SISC 36, 2014)
4. A priori assess $u \approx \tilde{u} \in HT(T, k)$
5. Postprocess $\tilde{u}$ to find max, min, cuts, ...
6. Prove (global) convergence of alternating minimization
7. From trees to networks
8. Relation between HT and deep convolutional neural networks

   ...

Thank You!

L. Grasedyck,
Hierarchical Singular Value Decomposition of Tensors,
SIMAX 31 (2010). `lgr@igpm.rwth-aachen.de`

W. Dahmen, R. DeVore, L. Grasedyck, E. Süli,
Tensor-Sparsity of Solutions to High-Dimensional Elliptic Partial Differential Equations,
FOCM 16 (2016). `dahmen@igpm.rwth-aachen.de`

L. Grasedyck, S. Krämer
Stable ALS Approximation in the TT-Format for Rank-Adaptive Tensor Completion,
arXiv:1701.08045 (2017). `kraemer@igpm.rwth-aachen.de`

supported by the DFG within priority program 1648, 1886