Tensor Numerical Methods

Fundamental principles and application to TEI calculations

Martin Mrovec

Katedra aplikované matematiky FEI VŠB-TU Ostrava, IT4Innovations

29. 3. 2016

0. Motivation

SVD

Consider a real rectangular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{v} \in \mathbb{R}^{n}$. Computation of matrix vector product

$$\mathbf{A}\mathbf{v} = \mathbf{x} \in \mathbb{R}^{m} \tag{1}$$

costs n^2 operations.

SVD

Consider a real rectangular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{v} \in \mathbb{R}^{n}$.
Computation of matrix vector product

$$\mathbf{A}\mathbf{v} = \mathbf{x} \in \mathbb{R}^{m} \tag{1}$$

costs n^2 operations.
If we know the **S**ingular **V**alue **D**ecomposition of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{T} \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$ are orthonormal matrices, $\mathbf{S} \in \mathbb{R}^{r \times r}$ is diagonal matrix and $r = \operatorname{rank}(\mathbf{A})$, we can rewrite (1) as

$$\mathbf{U}\mathbf{S}\mathbf{V}^{T}\mathbf{v} = \mathbf{x}. \tag{3}$$

SVD

WHY?

SVD

WHY?

For simplicity consider an orthogonal (generally not orthonormal) matrix $\tilde{\mathbf{V}} = \left(\mathbf{S}\mathbf{V}^T\right)^T$, so

$$\mathbf{A} = \mathbf{U}\tilde{\mathbf{V}}^T. \tag{4}$$

In case of low-rank matrix ($r \ll n$) the matrix vector multiplication using SVD is much cheaper:

SVD

WHY?

For simplicity consider an orthogonal (generally not orthonormal) matrix $\tilde{\mathbf{V}} = \left(\mathbf{S}\mathbf{V}^T\right)^T$, so

$$\mathbf{A} = \mathbf{U}\tilde{\mathbf{V}}^T. \tag{4}$$

In case of low-rank matrix ($r \ll n$) the matrix vector multiplication using SVD is much cheaper:

$$\mathbf{V}^T\mathbf{v} = \mathbf{z} \quad \dots \quad rn$$

SVD

WHY?

For simplicity consider an orthogonal (generally not orthonormal) matrix $\tilde{\mathbf{V}} = \left(\mathbf{S}\mathbf{V}^T\right)^T$, so

$$\mathbf{A} = \mathbf{U}\tilde{\mathbf{V}}^T. \tag{4}$$

In case of low-rank matrix ($r \ll n$) the matrix vector multiplication using SVD is much cheaper:

$$\begin{aligned} \mathbf{V}^T\mathbf{v} = \mathbf{z} \quad &\ldots \quad rn \\ \mathbf{U}\mathbf{z} = \mathbf{x} \quad &\ldots \quad rn \end{aligned}$$

SVD

WHY?

For simplicity consider an orthogonal (generally not orthonormal) matrix $\tilde{\mathbf{V}} = \left(\mathbf{S}\mathbf{V}^T\right)^T$, so

$$\mathbf{A} = \mathbf{U}\tilde{\mathbf{V}}^T. \tag{4}$$

In case of low-rank matrix ($r \ll n$) the matrix vector multiplication using SVD is much cheaper:

$$\begin{aligned} \mathbf{V}^T\mathbf{v} = \mathbf{z} \quad &\ldots \quad rn \\ \mathbf{U}\mathbf{z} = \mathbf{x} \quad &\ldots \quad rn \\ \hline \mathbf{U}\mathbf{V}^T\mathbf{v} = \mathbf{z} \quad &\ldots \quad 2rn \ll n^2 \end{aligned}$$

SVD

WHY?

For simplicity consider an orthogonal (generally not orthonormal) matrix $\tilde{\mathbf{V}} = \left(\mathbf{S}\mathbf{V}^T\right)^T$, so

$$\mathbf{A} = \mathbf{U}\tilde{\mathbf{V}}^T. \tag{4}$$

In case of low-rank matrix ($r \ll n$) the matrix vector multiplication using SVD is much cheaper:

$$\begin{aligned} \mathbf{V}^T\mathbf{v} = \mathbf{z} \quad &\ldots \quad rn \\ \mathbf{U}\mathbf{z} = \mathbf{x} \quad &\ldots \quad rn \\ \hline \mathbf{U}\mathbf{V}^T\mathbf{v} = \mathbf{z} \quad &\ldots \quad 2rn \ll n^2 \end{aligned}$$

However, rank r does not have to be low.

SVD

Solution = Truncated SVD Let $\mathbf{USV}^T$ be the SVD decomposition of $\mathbf{A}$. Without loss of generality consider a non-increasing sequence of singular values

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0, \quad (5)$$

so $\mathbf{S} = \operatorname{diag}([\sigma_1, \sigma_2, \ldots, \sigma_n])$. Set

$$\mathbf{S}_k = \operatorname{diag}([\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0]) \in \mathbb{R}^{n \times n} \quad (6)$$

$$\overline{\mathbf{S}}_k = \operatorname{diag}([\sigma_1, \sigma_2, \ldots, \sigma_k]) \in \mathbb{R}^{k \times k} \quad (7)$$

and matrices $\overline{\mathbf{U}}_k, \overline{\mathbf{V}}_k$ as first k columns of matrices $\mathbf{U}, \mathbf{V}$. Define

$$\mathbf{A}_k := \mathbf{U}\mathbf{S}_k\mathbf{V}^T = \overline{\mathbf{U}}_k\overline{\mathbf{S}}_k\overline{\mathbf{V}}_k^T. \quad (8)$$

SVD

$$\mathbf{A}_k := \mathbf{U}\mathbf{S}_k\mathbf{V}^T = \overline{\mathbf{U}}_k\overline{\mathbf{S}}_k\overline{\mathbf{V}}_k^T.$$

Then $\mathbf{A}_k$ is the best rank-k approximation of $\mathbf{A}$, i.e.

$$\|\mathbf{A} - \mathbf{B}\| \geq \|\mathbf{A} - \mathbf{A}_k\|, \ \text{rank}\,(\mathbf{B}) \leq k. \tag{9}$$

(Eckart-Young Theorem).
Error of approximation:

$$\|\mathbf{A} - \mathbf{A}_k\|_F = \sqrt{\sum_{i=k+1}^{n} \sigma_i^2} \tag{10}$$

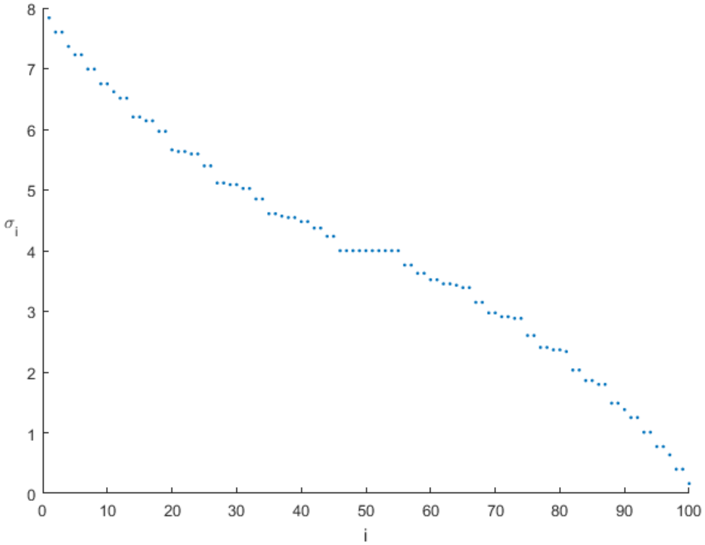

Figure: Singular values of discretized 2D laplacian (matrix 100 $\times$ 100).

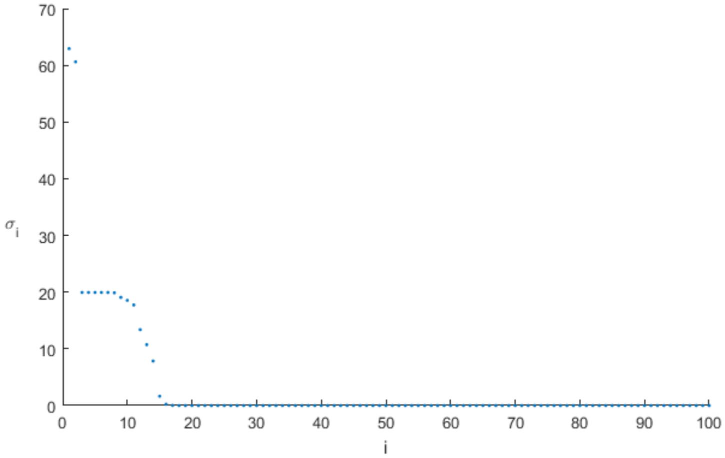

Figure: Singular values of discretized function $f(x, y) = \sin(x + y)\cos(xy) - y^2\cos(x)$ on interval $[0, 2\pi] \times [0, 2\pi]$ (matrix 100×100).

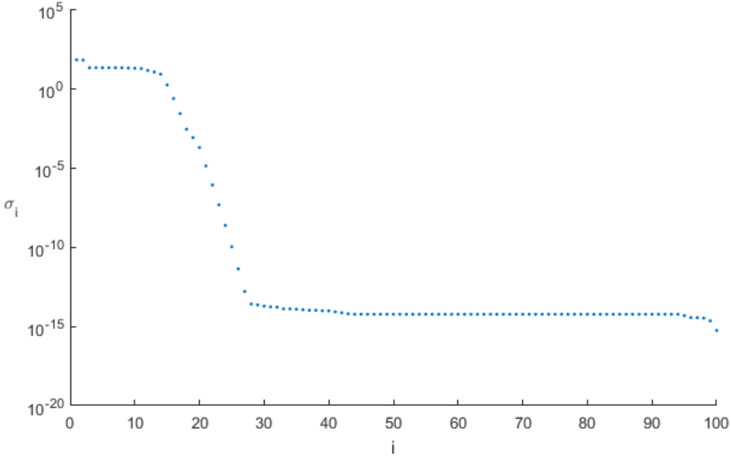

Figure: Singular values of discretized function $f(x,y) = \sin(x+y)\cos(xy) - y^2\cos(x)$ on interval $[0, 2\pi] \times [0, 2\pi]$ (matrix 100×100). Logarithmic scale.

Tensor

Definition

A tensor of order d (N-d tensor) is a multidimensional array over a d-tuple index set,

$$\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}, \ [\mathbf{A}]_{i_1 i_2 \ldots i_d} = a_{i_1 i_2 \ldots i_d} \in \mathbb{R}. \quad (11)$$

Tensor

Definition

A tensor of order d (N-d tensor) is a multidimensional array over a d-tuple index set,

$$\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}, \ [\mathbf{A}]_{i_1 i_2 \ldots i_d} = a_{i_1 i_2 \ldots i_d} \in \mathbb{R}. \tag{11}$$

An N-d tensor $=$ an element of a tensor-product Hilbert space (finite dimensional)

$$\mathbb{W}_{\mathbf{n}} \equiv \mathbb{W}_{\mathbf{n},d} = \bigotimes_{\ell=1}^{d} X_\ell \tag{12}$$

where

$$X_\ell = \mathbb{R}^{n_\ell}, \ \mathbf{n} = (n_1, \ldots, n_d)\,. \tag{13}$$

Scalar (inner) product is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle := \sum_{\mathbf{i} \in \mathcal{I}} [\mathbf{A}]_{\mathbf{i}} [\mathbf{B}]_{\mathbf{i}}, \quad \mathbf{A}, \mathbf{B} \in \mathbb{W}_{\mathbf{n}}. \tag{14}$$

Induced norm:

$$\|\mathbf{A}\| := \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle} = \sqrt{\sum_{\mathbf{i} \in \mathcal{I}} [\mathbf{A}]_{\mathbf{i}}^2}. \tag{15}$$

(Analogy to Frobenius norm)

Multilinear algebra operations

Hadamard product, Discrete convolution

Hadamard (entrywise) product $\odot : \mathbb{W}_{\mathbf{n}} \times \mathbb{W}_{\mathbf{n}} \to \mathbb{W}_{\mathbf{n}}$ is defined as

$$[\mathbf{A} \odot \mathbf{B}]_{\mathbf{i}} = [\mathbf{A}]_{\mathbf{i}} [\mathbf{B}]_{\mathbf{i}}, \ \mathbf{i} \in \mathcal{I}. \tag{16}$$

Multilinear algebra operations
Hadamard product, Discrete convolution

Hadamard (entrywise) product $\odot : \mathbb{W}_{\mathbf{n}} \times \mathbb{W}_{\mathbf{n}} \to \mathbb{W}_{\mathbf{n}}$ is defined as

$$[\mathbf{A} \odot \mathbf{B}]_{\mathbf{i}} = [\mathbf{A}]_{\mathbf{i}} [\mathbf{B}]_{\mathbf{i}}, \ \mathbf{i} \in \mathcal{I}. \tag{16}$$

For given tensors **A** and **B** the discrete convolution can be defined entrywise as

$$[\mathbf{A} * \mathbf{B}]_{\mathbf{j}} = \left[\sum_{\mathbf{k} \in \mathcal{I}} [\mathbf{A}]_{\mathbf{k}} [\mathbf{B}]_{\mathbf{j}-\mathbf{k}} \right]_{\mathbf{j}},$$

$$\mathbf{j} \in \underbrace{\{1, 2, \ldots, 2n_1 - 1\} \times \ldots \times \{1, 2, \ldots, 2n_d - 1\}}_{=:\mathcal{J}}. \tag{17}$$

Operations are very expensive for large d:

- scalar product: $\mathcal{O}\left(n^d\right)$
- Hadamard product: $\mathcal{O}\left(n^d\right)$
- discrete convolution: $\mathcal{O}\left(n^{2d}\right)$ ($n^d \log n$ if we use the fast FFT algorithm)

Operations are very expensive for large d:

- scalar product: $\mathcal{O}\left(n^d\right)$
- Hadamard product: $\mathcal{O}\left(n^d\right)$
- discrete convolution: $\mathcal{O}\left(n^{2d}\right)$ ($n^d \log n$ if we use the fast FFT algorithm)

Storage cost: $\mathcal{O}\left(n^d\right)$

Operations are very expensive for large d:

- scalar product: $\mathcal{O}\left(n^d\right)$
- Hadamard product: $\mathcal{O}\left(n^d\right)$
- discrete convolution: $\mathcal{O}\left(n^{2d}\right)$ ($n^d \log n$ if we use the fast FFT algorithm)

Storage cost: $\mathcal{O}\left(n^d\right)$

"The Curse of Dimensionality".

Contracted product

Given a tensor $\mathbf{A} \in \mathbb{R}^{n_1,\dots,n_{\ell-1},n_\ell,n_{\ell+1},\dots,n_d}$ and a matrix $\mathbf{B} \in \mathbb{R}^{n_\ell,m}$, we define a contracted product along mode ℓ as

$$\mathbb{R}^{n_1,\dots,n_{\ell-1},m,n_{\ell+1},\dots,n_d} \ni \mathbf{C} = \mathbf{A} \times_\ell \mathbf{B} \tag{18}$$

where each entry of the tensor $\mathbf{C}$ may be written as

$$c_{i_1,\dots,i_{\ell-1},j_\ell,i_{\ell+1},\dots,i_d} = \sum_{k=1}^{n_\ell} a_{i_1,\dots,i_{\ell-1},k,i_{\ell+1},\dots,i_d} b_{k,j_\ell}, \; j_\ell = 1,\dots,m. \tag{19}$$

Consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. The vector representation $vec\left(\mathbf{A}\right) \in \mathbf{R}^{mn} =$ putting the matrix columns (ordered) into the vector

$$vec\left(\mathbf{A}\right) = \left[a_{11}, \ldots, a_{m1}, a_{12}, \ldots, a_{m2}, \ldots \quad \ldots, a_{1n}, \ldots, a_{mm}\right]^T, \qquad (20)$$

or

$$vec\left(\mathbf{A}\right) = \begin{bmatrix} \left[a\left(:,1\right)\right] \\ \left[a\left(:,2\right)\right] \\ \vdots \\ \left[a\left(:,n\right)\right] \end{bmatrix}. \qquad (21)$$

In case of arbitrary tensor $\mathbf{B} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ we can define it recursively:

$$vec\left(\mathbf{B}\right) = \begin{bmatrix} vec\left(\left[b\left(:, \dots, :, 1\right)\right]\right) \\ vec\left(\left[b\left(:, \dots, :, 2\right)\right]\right) \\ \vdots \\ vec\left(\left[b\left(:, \dots, :, n\right)\right]\right) \end{bmatrix}. \qquad (22)$$

"Stacking" of 1-mode fibers of **B**.
Index mapping:

$$j = 1 + \sum_{k=1}^{d} \left(i_k - 1\right) \prod_{\ell=1}^{k-1} n_\ell. \qquad (23)$$

Tensor reshaping
Matrix unfolding

Matrix unfolding of a given tensor $\mathbf{B} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ along mode ℓ is a matrix $\mathbf{B}_{(\ell)} \in \mathbb{R}^{n_\ell \times (n_1 \dots n_{\ell-1} n_{\ell+1} \dots n_d)}$ which columns are ordered ℓ-mode fibers of $\mathbf{B}$.

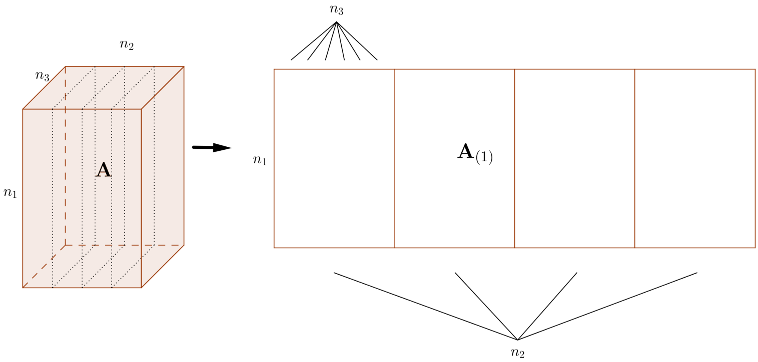

Figure: Matrix unfolding of 3-d tensor **A** along mode 1

2. Problem setting

Calculation of TEI

To solve discretized Hartree-Fock equations (e.g. using the Direct Inversion of the Iterative Subspace (DIIS) or Inexact Restoration method (IRM)) we have to precompute so-called two-electron integrals:

$$b_{ijkl} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{g_i(\mathbf{x})\, g_j(\mathbf{x})\, g_k(\mathbf{y})\, g_l(\mathbf{y})}{\|\mathbf{x} - \mathbf{y}\|}\, \mathrm{d}^3\mathbf{y}\, \mathrm{d}^3\mathbf{x}, \tag{24}$$

where basis functions $g_* : \mathbb{R}^3 \to \mathbb{R}$ satisfy $|g_*(\mathbf{x})| \to 0$ when $\|\mathbf{x}\| \to \infty$.

Calculation of TEI

To solve discretized Hartree-Fock equations (e.g. using the Direct Inversion of the Iterative Subspace (DIIS) or Inexact Restoration method (IRM)) we have to precompute so-called two-electron integrals:

$$b_{ijkl} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{g_i(\mathbf{x})\, g_j(\mathbf{x})\, g_k(\mathbf{y})\, g_l(\mathbf{y})}{\|\mathbf{x}-\mathbf{y}\|}\, \mathrm{d}^3\mathbf{y}\, \mathrm{d}^3\mathbf{x}, \tag{24}$$

where basis functions $g_* : \mathbb{R}^3 \to \mathbb{R}$ satisfy $|g_*(\mathbf{x})| \to 0$ when $\|\mathbf{x}\| \to \infty$.

For gaussian basis functions

$$g(\mathbf{x}) = Np(\mathbf{x}) e^{-\alpha\|\mathbf{x}\|^2}, \tag{25}$$

where $p(\mathbf{x}) = p(x_1, x_2, x_3) = x_1^k x_2^l x_3^m,\ k, l, m \in \mathbb{N}_0$, an analytical solution is known.

Calculation of TEI

Generally an analytical solution is not known :-(.

Calculation of TEI

Generally an analytical solution is not known :-(.

Consider a finite box (e.g. a cube) with an equidistant discretization with step size h. So each basis function is approximated by 3-d tensor

$$g_* (\mathbf{x}) \approx \mathbf{G}_* \in \mathbb{R}^{n^{\otimes 3}}. \tag{26}$$

If we rewrite the TEI as

$$b_{ijkl} = \int_{\mathbb{R}^3} g_i (\mathbf{x}) \, g_j (\mathbf{x}) \int_{\mathbb{R}^3} g_k (\mathbf{y}) \, g_l (\mathbf{y}) \frac{1}{\|\mathbf{x} - \mathbf{y}\|} \, \mathrm{d}^3\mathbf{y} \, \mathrm{d}^3\mathbf{x}, \tag{27}$$

we can understand the inner integral as function of $\mathbf{x}$.

Calculation of TEI

The discretized problem:

$$b_{ijkl} \approx h^6 \langle \mathbf{G}_i \odot \mathbf{G}_j, \mathcal{P}_\mathcal{N} * (\mathbf{G}_k \odot \mathbf{G}_l) \rangle , \qquad (28)$$

where $\mathcal{P}_\mathcal{N} \in \mathbb{R}^{(2n-1)^{\otimes 3}}$ is discretized Newton convolving kernel $\frac{1}{\|\mathbf{y}\|}$. To resolve singularities we have to consider very fine grid - in order of 10^5 - 10^6 in one dimension.

Calculation of TEI

The discretized problem:

$$b_{ijkl} \approx h^6 \left\langle \mathbf{G}_i \odot \mathbf{G}_j, \mathcal{P}_{\mathcal{N}} * (\mathbf{G}_k \odot \mathbf{G}_l) \right\rangle, \tag{28}$$

where $\mathcal{P}_{\mathcal{N}} \in \mathbb{R}^{(2n-1)^{\otimes 3}}$ is discretized Newton convolving kernel $\frac{1}{\|\mathbf{y}\|}$. To resolve singularities we have to consider very fine grid - in order of 10^5 - 10^6 in one dimension.

Tensor $\mathbf{G}_*$ has 10^{15} entries!!

Calculation of TEI

The discretized problem:

$$b_{ijkl} \approx h^6 \left\langle \mathbf{G}_i \odot \mathbf{G}_j, \mathcal{P}_{\mathcal{N}} * (\mathbf{G}_k \odot \mathbf{G}_l) \right\rangle , \tag{28}$$

where $\mathcal{P}_{\mathcal{N}} \in \mathbb{R}^{(2n-1)^{\otimes 3}}$ is discretized Newton convolving kernel $\frac{1}{\|\mathbf{y}\|}$. To resolve singularities we have to consider very fine grid - in order of 10^5 - 10^6 in one dimension.

Tensor $\mathbf{G}_*$ has 10^{15} entries!!

Solution - low rank tensor approximations.

3. Tensor formats

Tensor formats

Does there exist any decomposition of given tensor that corresponds to matrix SVD?

Tensor formats

Does there exist any decomposition of given tensor that corresponds to matrix SVD?

Not directly.

Tensor formats

Does there exist any decomposition of given tensor that corresponds to matrix SVD?

Not directly.

- Sum of rank 1 updates, no orthogonality - **Canonical format**

Tensor formats

Does there exist any decomposition of given tensor that corresponds to matrix SVD?

Not directly.

- Sum of rank 1 updates, no orthogonality - **Canonical format**
- Orthonormality, no diagonal core - **Tucker format**

Canonical format

A so-called rank-1 tensor $\mathbf{V} \in \mathbb{W}_{\mathbf{n}}$ can be written as a tensor product

$$\mathbf{V} = \mathbf{v}^{(1)} \otimes \ldots \otimes \mathbf{v}^{(d)} = \bigotimes_{\ell=1}^{d} \mathbf{v}^{(\ell)}, \quad \mathbf{v}^{(\ell)} \in \mathbb{R}^{n_\ell}. \tag{29}$$

Canonical format

A so-called rank-1 tensor $\mathbf{V} \in \mathbb{W}_{\mathbf{n}}$ can be written as a tensor product

$$\mathbf{V} = \mathbf{v}^{(1)} \otimes \ldots \otimes \mathbf{v}^{(d)} = \bigotimes_{\ell=1}^{d} \mathbf{v}^{(\ell)}, \quad \mathbf{v}^{(\ell)} \in \mathbb{R}^{n_\ell}. \tag{29}$$

Each tensor $\mathbf{A} \in \mathbb{W}_{\mathbf{n}}$ may be written as

$$\mathbf{A} = \sum_{k=1}^{R} \mathbf{a}_k^{(1)} \otimes \ldots \otimes \mathbf{a}_k^{(d)}, \quad \mathbf{a}_k^{(\ell)} \in \mathbb{R}^{n_\ell}. \tag{30}$$

Canonical format

A so-called rank-1 tensor $\mathbf{V} \in \mathbb{W}_{\mathbf{n}}$ can be written as a tensor product

$$\mathbf{V} = \mathbf{v}^{(1)} \otimes \ldots \otimes \mathbf{v}^{(d)} = \bigotimes_{\ell=1}^{d} \mathbf{v}^{(\ell)}, \quad \mathbf{v}^{(\ell)} \in \mathbb{R}^{n_\ell}. \tag{29}$$

Each tensor $\mathbf{A} \in \mathbb{W}_{\mathbf{n}}$ may be written as

$$\mathbf{A} = \sum_{k=1}^{R} \mathbf{a}_k^{(1)} \otimes \ldots \otimes \mathbf{a}_k^{(d)}, \quad \mathbf{a}_k^{(\ell)} \in \mathbb{R}^{n_\ell}. \tag{30}$$

That is so called Canonical R-term representation (we write $\mathbf{A} \in \mathcal{C}_R(\mathbb{W}_{\mathbf{n}})$ or $\mathbf{A} \in \mathcal{C}_{R,\mathbf{n}}$ which means that $\operatorname{rank}(\mathbf{A}) \leq R$). Minimal $R \in \mathbb{N}$ - canonical rank. It satisfies

$$1 \leq R \leq n^{d-1}. \tag{31}$$

Canonical format $\mathcal{C}_{R,\mathbf{n}}$

$\mathbf{i}$-th entry of $\mathbf{A}$, we simply evaluate

$$[\mathbf{A}]_{\mathbf{i}} = \sum_{k=1}^{R} \left[\mathbf{a}_k^{(1)}\right]_{i_1} \cdot \ldots \cdot \left[\mathbf{a}_k^{(d)}\right]_{i_d}. \quad (32)$$

Canonical format $\mathcal{C}_{R,\mathbf{n}}$

$\mathbf{i}$-th entry of $\mathbf{A}$, we simply evaluate

$$[\mathbf{A}]_{\mathbf{i}} = \sum_{k=1}^{R} \left[\mathbf{a}_k^{(1)}\right]_{i_1} \cdot \ldots \cdot \left[\mathbf{a}_k^{(d)}\right]_{i_d} . \tag{32}$$

Sometimes we require vectors $\mathbf{a}_k^{(\ell)}$ to be normalized. Then we substitute

$$\mathbf{a}_k^{(\ell)} = c_k^{(\ell)} \mathbf{u}_k^{(\ell)}, \; c_k^{(\ell)} \in \mathbb{R}, \; \left\| \mathbf{u}_k^{(\ell)} \right\| = 1. \tag{33}$$

So the equivalent normalized canonical representation is written as

$$\mathbf{A} = \sum_{k=1}^{R} c_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)}, \tag{34}$$

c_k coefficients are similar to singular values - possible rank reduction.

Canonical format $\mathcal{C}_{R,\mathbf{n}}$

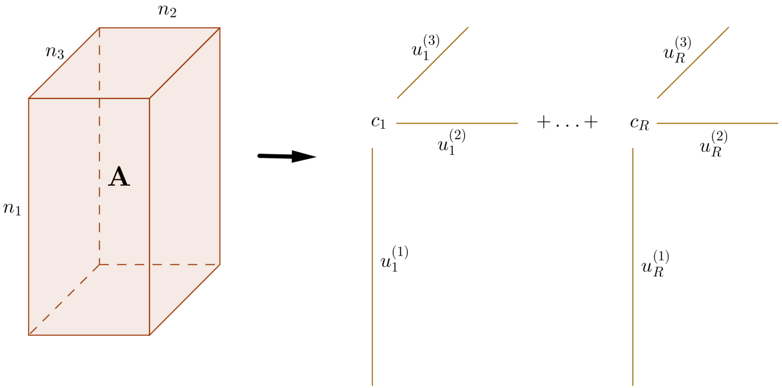

Figure: Canonical representation of tensor **A**

Canonical format $\mathcal{C}_{R,\mathbf{n}}$

pros and cons

+ Storage cost dRn
+ Cheap tensor operations

Canonical format $\mathcal{C}_{R,\mathbf{n}}$

pros and cons

+ Storage cost dRn
+ Cheap tensor operations
- No algorithm to get canonical rank (set of "max R-rank tensors" not closed)
- No algorithm to find the best rank-R approximation (does not have to exist)

Tucker format

Given the rank parameter $\mathbf{r} = (r_1, \ldots, r_d)$ we define the orthogonal Tucker format as

$$\mathbf{A} := \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}} b_{k_1,\ldots,k_d} \mathbf{v}_{k_1}^{(1)} \otimes \ldots \otimes \mathbf{v}_{k_d}^{(d)} \tag{35}$$

where $\mathbf{k} = (k_1, \ldots, k_d)$ and $\forall \ell \in \{1, \ldots, d\}$ the set of vectors $\mathbf{v}_{k_\ell}^{(\ell)}$, $1 \leq k_\ell \leq r_\ell$ is orthonormal. Coefficients $b_{k_1,\ldots,k_d}$ form a tensor $\mathbf{B}$ of order d. We write $\mathbf{A} \in \mathcal{T}_{\mathbf{r}}(\mathbb{W}_{\mathbf{n}})$ or $\mathbf{A} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$.

Tucker format

Given the rank parameter $\mathbf{r} = (r_1, \ldots, r_d)$ we define the orthogonal Tucker format as

$$\mathbf{A} := \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}} b_{k_1,\ldots,k_d} \mathbf{v}_{k_1}^{(1)} \otimes \ldots \otimes \mathbf{v}_{k_d}^{(d)} \tag{35}$$

where $\mathbf{k} = (k_1, \ldots, k_d)$ and $\forall \ell \in \{1, \ldots, d\}$ the set of vectors $\mathbf{v}_{k_\ell}^{(\ell)},\ 1 \leq k_\ell \leq r_\ell$ is orthonormal. Coefficients $b_{k_1,\ldots,k_d}$ form a tensor $\mathbf{B}$ of order d. We write $\mathbf{A} \in \mathcal{T}_{\mathbf{r}}(\mathbb{W}_{\mathbf{n}})$ or $\mathbf{A} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$.

Contracted product notation:

$$\mathbf{A} = \mathbf{B} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)} \tag{36}$$

Tucker format

Given the rank parameter $\mathbf{r} = (r_1, \ldots, r_d)$ we define the orthogonal Tucker format as

$$\mathbf{A} := \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}} b_{k_1,\ldots,k_d} \mathbf{v}_{k_1}^{(1)} \otimes \ldots \otimes \mathbf{v}_{k_d}^{(d)} \tag{35}$$

where $\mathbf{k} = (k_1, \ldots, k_d)$ and $\forall \ell \in \{1, \ldots, d\}$ the set of vectors $\mathbf{v}_{k_\ell}^{(\ell)}$, $1 \leq k_\ell \leq r_\ell$ is orthonormal. Coefficients $b_{k_1,\ldots,k_d}$ form a tensor $\mathbf{B}$ of order d. We write $\mathbf{A} \in \mathcal{T}_{\mathbf{r}}(\mathbb{W}_{\mathbf{n}})$ or $\mathbf{A} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$.

Contracted product notation:

$$\mathbf{A} = \mathbf{B} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)} \tag{36}$$

$\mathbf{i}$-th element of the tensor $\mathbf{A}_{(\mathbf{r})}$:

$$[\mathbf{A}]_{\mathbf{i}} = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}} b_{k_1,\ldots,k_d} \left[\mathbf{v}_{k_1}^{(1)}\right]_{i_1} \cdot \ldots \cdot \left[\mathbf{v}_{k_d}^{(d)}\right]_{i_d}. \tag{37}$$

Tucker format $\mathcal{T}_{\mathbf{r},\mathbf{n}}$

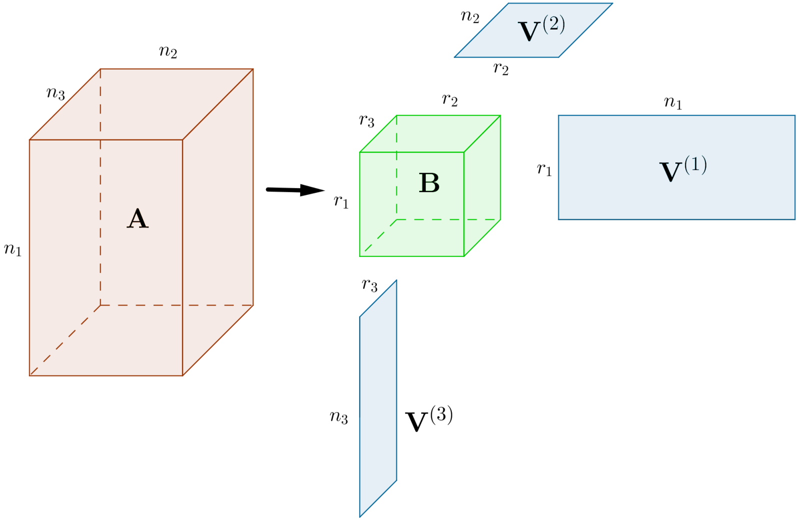

Figure: Tensor **A** in Tucker format

Theorem

(higher order SVD - HOSVD)
Each tensor $\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *can be written as the product*

$$\mathbf{A} = \mathbf{S} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}, \tag{38}$$

where $\mathbb{R}^{n_\ell \times n_\ell} \ni \mathbf{V}^{(\ell)} = \left[\mathbf{v}_1^{(\ell)} \mathbf{v}_2^{(\ell)} \ldots \mathbf{v}_{n_\ell}^{(\ell)}\right]$ *are orthonormal matrices* $\left(\left(\mathbf{V}^{(\ell)}\right)^T \mathbf{V}^{(\ell)} = I \in \mathbb{R}^{n_\ell \times n_\ell}\right)$ *and the tensor* $\mathbf{S}$ *has the same size as* $\mathbf{A}$. *For fixed index* ℓ *sub-tensors of* $\mathbf{S}$, $\mathbf{S}_{i_\ell = \alpha}$, *have the following properties*

1. $\left\langle \mathbf{S}_{i_\ell=\alpha}, \mathbf{S}_{i_\ell=\beta} \right\rangle = 0$ *when* $\alpha \neq \beta$ *(all-orthogonality),*
2. $\left\|\mathbf{S}_{i_\ell=1}\right\| \geq \left\|\mathbf{S}_{i_\ell=2}\right\| \geq \ldots \geq \left\|\mathbf{S}_{i_\ell=n_\ell}\right\| \geq 0$ *(ordering).*
3. $\mathbf{S} = \mathbf{A} \times_1 \left(\mathbf{V}^{(1)}\right)^T \times_2 \ldots \times_d \left(\mathbf{V}^{(d)}\right)^T$, *(*$\mathbf{S}$ *is the projection of A onto the tensor basis formed by orthonormal matrices* $\mathbf{V}^{(\ell)}$*)*

Frobenius norms $\left\|\mathbf{S}_{i_\ell=j}\right\|$ *are equal to the singular values* $\sigma_j^{(\ell)}$ *of the matrix unfolding* $\mathbf{A}_{(\ell)}$ *(*ℓ*-mode singular values). Vectors* $\mathbf{v}_j^{(\ell)}$ *are corresponding left singular vectors of* $\mathbf{A}_{(\ell)}$.

Algorithm

(HOSVD)

1. Given a tensor $\mathbf{A}$, for each $\ell = 1, \ldots, d$ construct the matrix unfolding $\mathbf{A}_{(\ell)}$.
2. For each unfolding find an SVD decomposition, i.e. $\mathbf{A}_{(\ell)} = \mathbf{V}^{(\ell)} \Sigma^{(\ell)} \left(\mathbf{W}^{(\ell)}\right)^T$. Store the matrices $\mathbf{V}^{(\ell)}$.
3. Compute the core tensor as $\mathbf{S} = \mathbf{A} \times_1 \left(\mathbf{V}^{(1)}\right)^T \times_2 \ldots \times_d \left(\mathbf{V}^{(d)}\right)^T$.

Theorem

(Truncated HOSVD)
Let the HOSVD of the tensor $\mathbf{A}$ *is given and let the rank of matrix unfolding* $\mathbf{A}_{(\ell)}$ *be equal to* R_ℓ *for* $\ell = 1, \dots, d$. *Let's denote by* $\tilde{\mathbf{A}}$ *the tensor which is obtained from* $\mathbf{A}$ *by discarding the smallest* ℓ*-mode singular values* $\sigma^{(\ell)}_{r_\ell+1}, \sigma^{(\ell)}_{r_\ell+2}, \dots, \sigma^{(\ell)}_{R_\ell}$ *for given values of* r_ℓ $(\ell = 1, \dots, d)$. *Then the error of the approximation satisfies*

$$\left\| \mathbf{A} - \tilde{\mathbf{A}} \right\|^2 \leq \sum_{\ell=1}^{d} \sum_{i_\ell = r_\ell+1}^{R_\ell} \left(\sigma^{(\ell)}_{i_\ell} \right)^2 . \tag{39}$$

Tucker format
truncated HOSVD

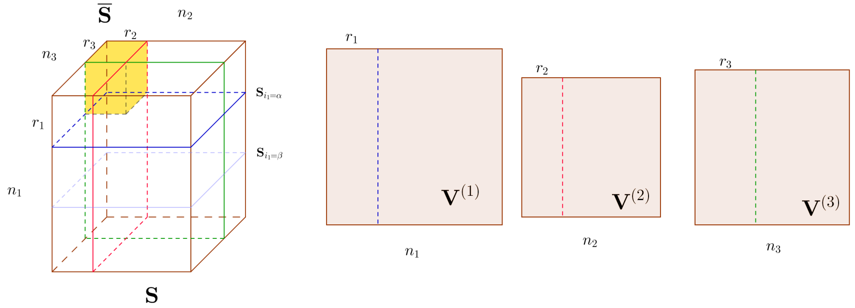

Figure: Mechanism of truncated HOSVD

Tucker format

pros and cons

- + Storage cost $r^d + drn$
- + Cheap tensor operations (more expensive than in case of can. format)
- + Quite robust algorithms to find the best rank-$\mathbf{r}$ approximation
- + Efficient especially for 3D tensors

Tucker format
pros and cons

- \+ Storage cost $r^d + drn$
- \+ Cheap tensor operations (more expensive than in case of can. format)
- \+ Quite robust algorithms to find the best rank-$\mathbf{r}$ approximation
- \+ Efficient especially for 3D tensors
- \- Expensive for larger d

Two-level Tucker-canonical format

Definition

Given rank parameters $R, \mathbf{r}$ we denote by $\mathcal{T}_{\mathcal{C}_{R,\mathbf{r}}}$ a subclass of tensors written in Tucker format $\mathcal{T}_{\mathbf{r},\mathbf{n}}$ whose core tensor $\mathbf{B}$ is represented in the canonical format.

Two-level Tucker-canonical format

Definition

Given rank parameters $R, \mathbf{r}$ we denote by $\mathcal{T}_{\mathcal{C}_{R,\mathbf{r}}}$ a subclass of tensors written in Tucker format $\mathcal{T}_{\mathbf{r},\mathbf{n}}$ whose core tensor $\mathbf{B}$ is represented in the canonical format.

A tensor in two-level format can be written as

$$\mathbf{A} = \left(\sum_{k=1}^{R} \xi_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)} \right) \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}. \qquad (40)$$

Two-level Tucker-canonical format

Definition

Given rank parameters $R, \mathbf{r}$ we denote by $\mathcal{T}_{\mathcal{C}_{R,\mathbf{r}}}$ a subclass of tensors written in Tucker format $\mathcal{T}_{\mathbf{r},\mathbf{n}}$ whose core tensor $\mathbf{B}$ is represented in the canonical format.

A tensor in two-level format can be written as

$$\mathbf{A} = \left(\sum_{k=1}^{R} \xi_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)} \right) \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}. \qquad (40)$$

- storage cost $dRr + drn$

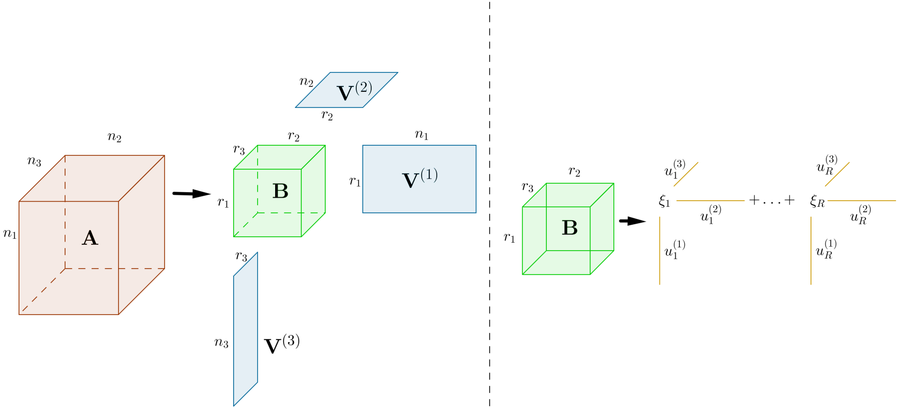

Figure: 3-d tensor **A** in two-level Tucker-canonical format

$\mathbf{A}, \mathbf{B} \in \mathcal{C}_{R,\mathbf{n}}$

$$\mathbf{A} = \sum_{k=1}^{R_A} c_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)}, \; \mathbf{B} = \sum_{j=1}^{R_B} d_j \mathbf{v}_j^{(1)} \otimes \ldots \otimes \mathbf{v}_j^{(d)}. \tag{41}$$

$\mathbf{A}, \mathbf{B} \in \mathcal{C}_{R,\mathbf{n}}$

$$\mathbf{A} = \sum_{k=1}^{R_A} c_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)}, \quad \mathbf{B} = \sum_{j=1}^{R_B} d_j \mathbf{v}_j^{(1)} \otimes \ldots \otimes \mathbf{v}_j^{(d)}. \tag{41}$$

- scalar product:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{k=1}^{R_A} \sum_{j=1}^{R_B} c_k d_j \prod_{\ell=1}^{d} \left\langle \mathbf{u}_k^{(\ell)}, \mathbf{v}_j^{(\ell)} \right\rangle \tag{42}$$

($dR_A R_B n$ operations)

$\mathbf{A}, \mathbf{B} \in \mathcal{C}_{R,\mathbf{n}}$

$$\mathbf{A} = \sum_{k=1}^{R_A} c_k \mathbf{u}_k^{(1)} \otimes \ldots \otimes \mathbf{u}_k^{(d)}, \;\; \mathbf{B} = \sum_{j=1}^{R_B} d_j \mathbf{v}_j^{(1)} \otimes \ldots \otimes \mathbf{v}_j^{(d)}. \tag{41}$$

- scalar product:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{k=1}^{R_A} \sum_{j=1}^{R_B} c_k d_j \prod_{\ell=1}^{d} \left\langle \mathbf{u}_k^{(\ell)}, \mathbf{v}_j^{(\ell)} \right\rangle \tag{42}$$

  ($dR_A R_B n$ operations)
- Hadamard product:

$$\mathbf{A} \odot \mathbf{B} = \sum_{k=1}^{R_A} \sum_{j=1}^{R_B} c_k d_j \bigotimes_{\ell=1}^{d} \left(\mathbf{u}_k^{(\ell)} \odot \mathbf{v}_j^{(\ell)} \right) \tag{43}$$

  ($dR_A R_B n$ operations)

- discrete convolution:

$$\mathbf{A} * \mathbf{B} = \sum_{k=1}^{R_A} \sum_{j=1}^{R_B} c_k d_j \bigotimes_{\ell=1}^{d} \left(\mathbf{u}_k^{(\ell)} * \mathbf{v}_j^{(\ell)} \right) \tag{44}$$

($dR_A R_B n \log n$ operations using 1D FFT)

Multilinear algebra operations
Tucker format

$\mathbf{A}, \mathbf{B} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$:

$$\mathbf{A} = \mathbf{C} \times_1 \mathbf{U}^{(1)} \times_2 \ldots \times_d \mathbf{U}^{(d)}, \ \mathbf{B} = \mathbf{D} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}. \qquad (45)$$

Multilinear algebra operations

Tucker format

$\mathbf{A}, \mathbf{B} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$:

$$\mathbf{A} = \mathbf{C} \times_1 \mathbf{U}^{(1)} \times_2 \ldots \times_d \mathbf{U}^{(d)}, \ \mathbf{B} = \mathbf{D} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}. \qquad (45)$$

- scalar product:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}_A} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{r}_B} c_{k_1 \ldots k_d} d_{j_1 \ldots j_d} \prod_{\ell=1}^{d} \left\langle \mathbf{u}_{k_\ell}^{(\ell)}, \mathbf{v}_{j_\ell}^{(\ell)} \right\rangle. \qquad (46)$$

($\mathcal{O}\left(dr^2 n + r^{2d}\right)$ operations)

$\mathbf{A}, \mathbf{B} \in \mathcal{T}_{\mathbf{r},\mathbf{n}}$:

$$\mathbf{A} = \mathbf{C} \times_1 \mathbf{U}^{(1)} \times_2 \ldots \times_d \mathbf{U}^{(d)}, \ \mathbf{B} = \mathbf{D} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_d \mathbf{V}^{(d)}. \qquad (45)$$

- scalar product:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}_A} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{r}_B} c_{k_1 \ldots k_d} d_{j_1 \ldots j_d} \prod_{\ell=1}^{d} \left\langle \mathbf{u}_{k_\ell}^{(\ell)}, \mathbf{v}_{j_\ell}^{(\ell)} \right\rangle. \qquad (46)$$

($\mathcal{O}\left(dr^2 n + r^{2d}\right)$ operations)

- Hadamard product:

$$\mathbf{A} \odot \mathbf{B} = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}_A} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{r}_B} c_{k_1 \ldots k_d} d_{j_1 \ldots j_d} \bigotimes_{\ell=1}^{d} \left(\mathbf{u}_{k_\ell}^{(\ell)} \odot \mathbf{v}_{j_\ell}^{(\ell)} \right). \qquad (47)$$

($\mathcal{O}\left(dr^2 n + r^{2d}\right)$ operations)

- discrete convolution:

$$\mathbf{A} * \mathbf{B} = \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{r}_A} \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{r}_B} c_{k_1 \ldots k_d} d_{j_1 \ldots j_d} \bigotimes_{\ell=1}^{d} \left(\mathbf{u}_{k_\ell}^{(\ell)} * \mathbf{v}_{j_\ell}^{(\ell)} \right). \tag{48}$$

($\mathcal{O}\left(dr^2 n \log n + r^{2d}\right)$ operations using 1D FFT)

Multilinear algebra operations

Summary

Tensor format	Scalar product	Hadamard product	Discrete convolution
Full	n^d	n^d	$n^d \log n$
Canonical	$dR_A R_B n$	$dR_A R_B n$	$dR_A R_B n \log n$
Tucker	$\mathcal{O}\left(dr^2 n + r^{2d}\right)$	$\mathcal{O}\left(dr^2 n + r^{2d}\right)$	$\mathcal{O}\left(dr^2 n \log n + r^{2d}\right)$

Table: Computational complexities of various tensor formats

4. Low rank approximation of function related tensors

Approximation of function related tensors

How to find an rank-structured approximation of tensor $\mathbf{G} \in \mathbb{R}^{n^{\otimes d}}$ discretizing a multivariate function $g : \mathbb{R}^d \to \mathbb{R}$ on a fine grid?

Approximation of function related tensors

How to find an rank-structured approximation of tensor $\mathbf{G} \in \mathbb{R}^{n^{\otimes d}}$ discretizing a multivariate function $g : \mathbb{R}^d \to \mathbb{R}$ on a fine grid?

- Tucker format
- Canonical format

Consider a set $\{\xi_{\mathbf{j}}\}_{\mathbf{j}=\mathbf{1}}^{\mathbf{N}}$ of interpolation points, where $\mathbf{N} = (N_1, \ldots, N_d)$ represents a number of interpolation points at each dimension. The interpolant of the function g:

$$\mathcal{I}_{\mathbf{N}} g(\mathbf{x}) = \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{N}} g(\xi_{\mathbf{j}})\, l_{\mathbf{j}}(\mathbf{x}), \;\; l_{\mathbf{j}} : \mathbb{R}^d \to \mathbb{R} \tag{49}$$

such as

$$l_{\mathbf{j}}(\xi_{\mathbf{j}}) = 1, \;\; l_{\mathbf{j}}(\xi_{\mathbf{i}}) = 0 \;\mathbf{i} \neq \mathbf{j}. \tag{50}$$

Given orthonormal sets of interpolation functions

$$I_{\mathbf{j}} = \iota_{j_1}^{(1)} \otimes \ldots \otimes \iota_{j_d}^{(d)}, \tag{51}$$

we can write a Tucker approximation

$$g(\mathbf{x}) \approx \mathbf{G} = \Xi \times_1 \mathbf{I}^{(1)} \times_2 \ldots \times_d \mathbf{I}^{(d)} \in \mathcal{T}_{\mathbf{N},\mathbf{n}}, \tag{52}$$

where

$$[\Xi]_{\mathbf{j}} = g(\xi_{\mathbf{j}}) \tag{53}$$

and columns of side matrices $\mathbf{I}^{(\ell)} \in \mathbb{R}^{n_\ell \times N_\ell}$ are discretized functions $\iota_{j_\ell}^{(\ell)},\ j_\ell = 1, \ldots, N_\ell$.

Given orthonormal sets of interpolation functions

$$I_{\mathbf{j}} = \iota_{j_1}^{(1)} \otimes \ldots \otimes \iota_{j_d}^{(d)}, \tag{51}$$

we can write a Tucker approximation

$$g(\mathbf{x}) \approx \mathbf{G} = \Xi \times_1 \mathbf{I}^{(1)} \times_2 \ldots \times_d \mathbf{I}^{(d)} \in \mathcal{T}_{\mathbf{N},\mathbf{n}}, \tag{52}$$

where

$$[\Xi]_{\mathbf{j}} = g(\xi_{\mathbf{j}}) \tag{53}$$

and columns of side matrices $\mathbf{I}^{(\ell)} \in \mathbb{R}^{n_\ell \times N_\ell}$ are discretized functions $\iota_{j_\ell}^{(\ell)}$, $j_\ell = 1, \ldots, N_\ell$.

How to choose an orthonormal set of interpolating functions?

The sinc function is defined as

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, x \neq 0 \\ 1, \ x = 0 \end{cases} \tag{54}$$

The sinc function is defined as

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, x \neq 0 \\ 1, \ x = 0 \end{cases} \tag{54}$$

Functions representing band-limited signals can be exactly reconstructed by their sampling values. So if the support of Fourier transform of given function, $\hat{g}$, is subset of $\left[-\frac{\pi}{h}, \frac{\pi}{h}\right]$ $(g \in U_h \ )$, then

$$g(x) = \sum_{n=-\infty}^{\infty} g(nh)\, S_{n,h}(x), \tag{55}$$

where

$$S_{n,h}(x) = \operatorname{sinc}\left(\frac{x - nh}{h}\right). \tag{56}$$

The sinc function is defined as

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x}, x \neq 0 \\ 1, \ x = 0 \end{cases} \tag{54}$$

Functions representing band-limited signals can be exactly reconstructed by their sampling values. So if the support of Fourier transform of given function, $\hat{g}$, is subset of $\left[-\frac{\pi}{h}, \frac{\pi}{h}\right]$ ($g \in U_h$), then

$$g(x) = \sum_{n=-\infty}^{\infty} g(nh) S_{n,h}(x), \tag{55}$$

where

$$S_{n,h}(x) = \operatorname{sinc}\left(\frac{x - nh}{h}\right). \tag{56}$$

A set of functions $\{S_{n,h}(x)\}_{n \in \mathbb{Z}}$ forms an orthonormal basis of U_h.

Approximation of function related tensors

Canonical format

Functions that depend on the sum of single variables, respectively their squares, i.e.

$$g(\mathbf{x}) = g(\rho), \ \ \rho = \sum_{i=1}^{d} x_i, \ \text{resp.} \ \rho = \sum_{i=1}^{d} x_i^2. \tag{57}$$

Functions that depend on the sum of single variables, respectively their squares, i.e.

$$g(\mathbf{x}) = g(\rho), \ \rho = \sum_{i=1}^{d} x_i, \ \text{resp.} \ \rho = \sum_{i=1}^{d} x_i^2. \tag{57}$$

Assume that a function $g(\rho)$ is given by transformation

$$g(\rho) = \int_{\Omega} G(t) \, e^{\rho F(t)} \, \mathrm{d}t, \ \Omega \in \{\mathbb{R}, \mathbb{R}_+, [a, b]\}. \tag{58}$$

Functions that depend on the sum of single variables, respectively their squares, i.e.

$$g(\mathbf{x}) = g(\rho), \ \rho = \sum_{i=1}^{d} x_i, \ \text{resp. } \rho = \sum_{i=1}^{d} x_i^2. \tag{57}$$

Assume that a function $g(\rho)$ is given by transformation

$$g(\rho) = \int_{\Omega} G(t)\, e^{\rho F(t)}\, \mathrm{d}t, \ \Omega \in \{\mathbb{R}, \mathbb{R}_+, [a, b]\}. \tag{58}$$

Suitable quadrature

$$g(\rho) \approx \sum_{j=1}^{R} \omega_j G(t_j)\, e^{\rho F(t_j)} \tag{59}$$

where ω_j are quadrature heights, t_j are quadrature points.

Approximation of function related tensors

Canonical format

If $\rho = \sum_{i=1}^{d} x_i$ we can write

$$g\left(\mathbf{x}\right) \approx \sum_{j=1}^{R} \omega_j G\left(t_j\right) \prod_{i=1}^{d} e^{x_i F\left(t_j\right)}. \tag{60}$$

If $\rho = \sum_{i=1}^{d} x_i$ we can write

$$g(\mathbf{x}) \approx \sum_{j=1}^{R} \omega_j G(t_j) \prod_{i=1}^{d} e^{x_i F(t_j)}. \tag{60}$$

Fully separable representation!

If $\rho = \sum_{i=1}^{d} x_i$ we can write

$$g(\mathbf{x}) \approx \sum_{j=1}^{R} \omega_j G(t_j) \prod_{i=1}^{d} e^{x_i F(t_j)}. \tag{60}$$

Fully separable representation!

$$g(\mathbf{x}) \approx \sum_{j=1}^{R} c_j \mu_j^{(1)} \otimes \ldots \otimes \mu_j^{(d)}. \tag{61}$$

Sinc quadrature (rectangular rule):

$$\int_{\mathbb{R}} G(t) e^{\rho F(t)} \, \mathrm{d}t \approx \int_{\mathbb{R}} \sum_{n=-M}^{M} S_{n,h}(t) G(nh) e^{\rho F(nh)} \mathrm{d}t = \sum_{n=-M}^{M} G(nh) e^{\rho F(nh)} \tag{62}$$

$$f(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|} = \frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}}. \quad (63)$$

$$f(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|} = \frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}}. \tag{63}$$

Now consider a function $g(\rho) = \frac{1}{\rho}$. It can be represented using an integral transformation

$$\frac{1}{\rho} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-\rho^2 t^2} \, \mathrm{d}t. \tag{64}$$

$$f\left(\mathbf{x}\right) = \frac{1}{\|\mathbf{x}\|} = \frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}}. \tag{63}$$

Now consider a function $g\left(\rho\right) = \frac{1}{\rho}$. It can be represented using an integral transformation

$$\frac{1}{\rho} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-\rho^2 t^2} \, \mathrm{d}t. \tag{64}$$

For the Newton kernel we can write

$$\frac{1}{\|\mathbf{x}\|} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-t^2 \sum_{i=1}^3 x_i^2} \, \mathrm{d}t. \tag{65}$$

$$f(\mathbf{x}) = \frac{1}{\|\mathbf{x}\|} = \frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}}. \quad (63)$$

Now consider a function $g(\rho) = \frac{1}{\rho}$. It can be represented using an integral transformation

$$\frac{1}{\rho} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-\rho^2 t^2} \, \mathrm{d}t. \quad (64)$$

For the Newton kernel we can write

$$\frac{1}{\|\mathbf{x}\|} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-t^2 \sum_{i=1}^3 x_i^2} \, \mathrm{d}t. \quad (65)$$

After substitution:

$$\frac{1}{\rho} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}} \cosh w \frac{e^{-\rho^2 \log^2\left(1+e^{sinhw}\right)}}{1 + e^{-sinhw}} \, \mathrm{d}w. \quad (66)$$

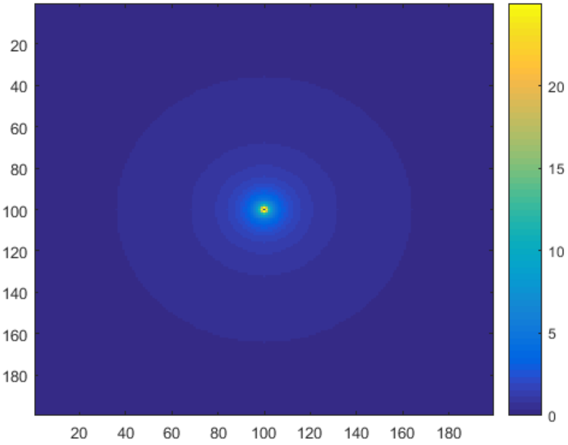

Figure: 2D newton potential discretized on square $[-2,2] \times [-2,2]$

Approximation of function related tensors

Canonical approximation of newton kernel

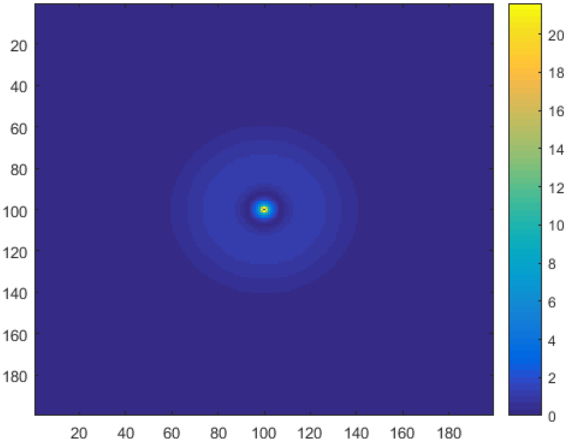

Figure: rank-3 approximation error

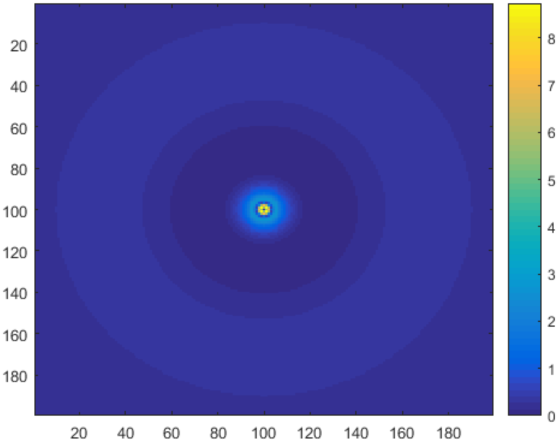

Figure: rank-5 approximation error

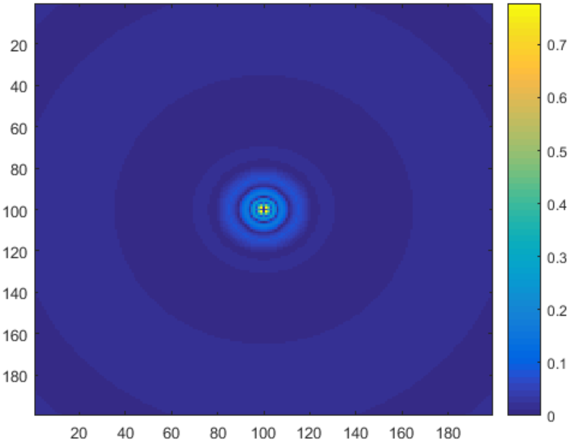

Figure: rank-11 approximation error

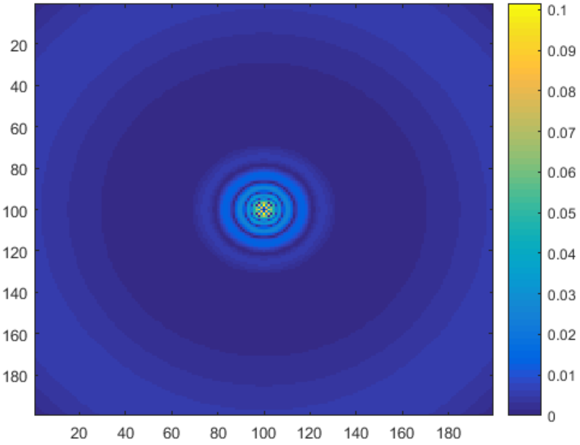

Figure: rank-15 approximation error

Other applications

Other applications

- tensor representation of multidimensional operators

Other applications

- tensor representation of multidimensional operators
- superfast FFT

Other applications

- tensor representation of multidimensional operators
- superfast FFT
- multi-parametric/stochastic PDEs

Other applications

- tensor representation of multidimensional operators
- superfast FFT
- multi-parametric/stochastic PDEs
- integrating of highly oscillating functions

Thank you for attention!

References

- Boris Khoromskij homepage: http://personal-homepages.mis.mpg.de/bokh/
- KHOROMSKIJ, Boris N. *Introduction to Tensor Numerical Methods in Scientific Computing* [online]. Zuerich, 2011 [cit. 2016-03-09]. Available at: http://www.math.uzh.ch/fileadmin/math/preprints/06_11.pdf
- KHOROMSKIJ, Boris N. *Tensor Numerical Methods for High-dimensional PDEs: Basic Theory and Initial Applications* [online]. Leipzig, 2014 [cit. 2016-03-09]. Available at: http://arxiv.org/pdf/1408.4053v1.pdf