CEMRACS,
July 19-23, 2021

Approximation and learning with tensor networks

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Tensor networks

Tensor networks are prominent tools for the representation or approximation of multivariate functions or multidimensional arrays.

- A long history in quantum physics.
- Tree tensor networks appeared independently in numerical analysis, as an extension of low-rank decompositions to high-order tensors.
- Growing use in statistics, data science and probabilistic modelling.

Computing with tensor networks

- For the approximation of a known tensor u with respect to a certain norm, we aim at finding a tensor network v with low complexity that minimizes

$$\|u - v\|.$$

  Here, the aim is the compression of u or the extraction of information from u (data analysis).

- For the solution of an equation $Au = b$ (e.g. in quantum physics, uncertainty quantification, stochastic calculus), we aim at finding a tensor network v with low complexity that minimizes some distance to u, e.g. some residual norm

$$\|Av - b\|.$$

  The aim is here to obtain an approximation of the solution u with a low computational complexity.

Computing with tensor networks

- In tensor completion, knowing some entries $(u(i))_{i\in\Omega}$ of a multidimensional array, we try to find a tensor network that suitably fit the data, e.g. by minimizing

$$\sum_{i\in\Omega} |u(i) - v(i)|^2,$$

The aim is here to recover (or complete) a tensor from partial information, by exploiting low-rank structures of the tensor.

- For inverse problems, we want to identify a tensor u from indirect and partial observations $y = Au$ or $y = Au + \epsilon$, where A is an observation map. We try to find a tensor network that suitably fit the observations by minimizing some distance

$$d(y, Av)$$

between observations and the prediction Av.
Exploiting low-rank structures in u allows to reduce the number of parameters to estimate and possibly makes the problem well-posed.

- Approximating a function u from evaluations $u(x^k)$ at some points x^k, e.g. by minimizing
$$\frac{1}{n}\sum_{k=1}^{n}(u(x^k) - v(x^k))^2.$$
Depending on the context, points can be given or chosen. Here we want to exploit at best the given evaluations or obtain a good approximation using a small number of evaluations.

Computing with tensor networks

- In supervised or unsupervised learning, tensor networks are used as a powerful model class for high-dimensional tasks.
- Supervised learning of the relation between a random variable Y and another random variable X. Introduction of a risk functional

$$\mathcal{R}(v) = \mathbb{E}(\ell(Y, v(X)))$$

that quantifies some expected distance between observations Y and predictions $v(X)$. In practice, using samples $\{(x_k, y_k)\}_{k=1}^n$, we optimize an empirical risk

$$\frac{1}{n}\sum_{k=1}^{n} \ell(y^k, v(x^k))$$

- Estimation of the density of a random variable X from samples $\{x_k\}_{k=1}^n$. If the density u minimizes some functional

$$\mathcal{R}(v) = \mathbb{E}(\gamma(v, X)),$$

we minimize in practice an empirical risk

$$\frac{1}{n}\sum_{k=1}^{n} \gamma(v, x^k)$$

Outline of the course

- Part I: Tensors, ranks and tensor networks
- Part II: Approximation theory of tree tensor networks
- Part III: Computational aspects

CEMRACS,
July 19-23, 2021

Approximation and learning with tensor networks

Part I: Tensors, ranks and tensor networks

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Outline

Outline

Algebraic tensors

Given d index sets $I_\nu = \{1, \dots, N_\nu\}$, $1 \leq \nu \leq d$, we introduce the multi-index set

$$I = I_1 \times \dots \times I_d.$$

An element v of the vector space $\mathbb{R}^I$ is a tensor of order d.

It can be represented by a multidimensional array

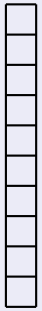
$$(v_i)_{i \in I} = (v_{i_1,\dots,i_d})_{i_1 \in I_1,\dots,i_d \in I_d}$$

that contains the coefficients of v in the canonical basis of $\mathbb{R}^I$, also denoted

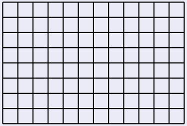
$$v(i) = v(i_1, \dots, i_d).$$

The order d is the number of dimensions, also known as ways or modes.

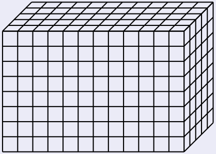
$d = 1$



$d = 2$

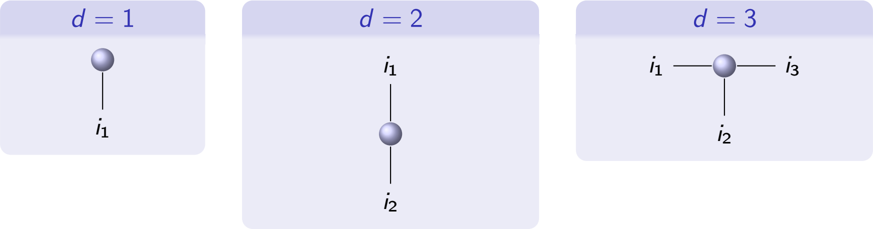


$d = 3$

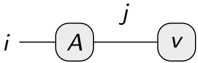


Tensor diagram notations

A tensor is represented by a solid shape and tensor indices are notated by lines emanating from this shape.



Connecting two index lines means contraction (or summation) over the corresponding indices.



$$= \sum_j A(i,j)v(j)$$

Algebraic tensors

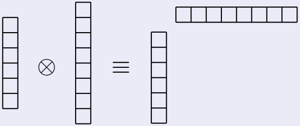
Given d vectors $v^{(\nu)} \in \mathbb{R}^{I_\nu}$, $1 \leq \nu \leq d$, the tensor product of these vectors

$$v := v^{(1)} \otimes \ldots \otimes v^{(d)}$$

is called an elementary tensor and is such that

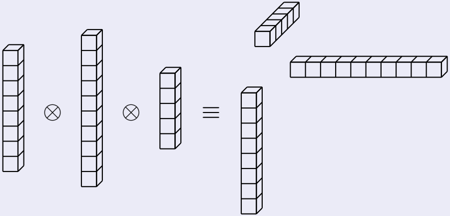
$$v(i) = v^{(1)}(i_1) \ldots v^{(d)}(i_d)$$

$d = 2$



Using matrix notations, $v \otimes w$ is identified with the matrix vw^T.

$d = 3$



Algebraic tensors

The tensor space $\mathbb{R}^I = \mathbb{R}^{I_1 \times \ldots \times I_d}$, also denoted $\mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_d}$, is defined by

$$\mathbb{R}^I = \mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_d} = \mathrm{span}\{v^{(1)} \otimes \ldots \otimes v^{(d)} : v^{(\nu)} \in \mathbb{R}^{I_\nu}, 1 \leq \nu \leq d\}$$

The canonical norm on $\mathbb{R}^I$, also called the Frobenius norm, is given by

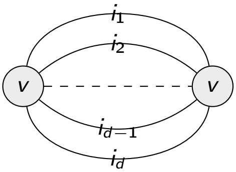
$$\|v\| = \sqrt{\sum_{i \in I} v(i)^2}$$

and makes $\mathbb{R}^I$ a Hilbert space. It coincides with the natural norm on $\ell_2(I)$. It is the only norm associated with an inner product and having the crossnorm property

$$\|v^{(1)} \otimes \ldots \otimes v^{(d)}\| = \|v^{(1)}\|_2 \ldots \|v^{(d)}\|_2.$$

In tensor diagram notations

$$\|v\|^2 = \sum_{i \in I} v(i)^2 =$$



Tensor product of functions

Let $V_\nu \subset \mathbb{R}^{\mathcal{X}_\nu}$ be a space of functions defined on $\mathcal{X}_\nu$.

$\mathcal{X}_\nu$ can be (a subset of) $\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$, $\mathbb{Z}$, or a set of vectors, sequences, graphs, images...

The tensor product of functions $v^{(\nu)} \in V_\nu$, denoted

$$v = v^{(1)} \otimes \ldots \otimes v^{(d)},$$

is a multivariate function defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ and such that

$$v(x_1, \ldots, x_d) = v^{(1)}(x_1) \ldots v^{(d)}(x_d)$$

Example

For $i \in \mathbb{N}_0^d$, the monomial $x^i = x_1^{i_1} \ldots x_d^{i_d}$ is an elementary tensor.

Tensor product of functions

The algebraic tensor product of spaces V_ν is defined as

$$V_1 \otimes \ldots \otimes V_d = \mathrm{span}\{v^{(1)} \otimes \ldots \otimes v^{(d)} : v^{(\nu)} \in V_\nu, 1 \leq \nu \leq d\}$$

which is the space of multivariate functions v which can be written as a finite linear combination of elementary (separated functions), i.e.

$$v(x_1, \ldots, x_d) = \sum_{k=1}^{n} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d).$$

Example

A polynomial $\sum_i a_i x^i$ with $x^i = x_1^{i_1} \ldots x_d^{i_d}$.

Up to a formal definition of the tensor product $\otimes$, the above construction can be extended to more general vector spaces (not only spaces of functions), including spaces of matrices or operators.

Infinite dimensional tensor spaces

For infinite dimensional spaces V_ν, a Hilbert (or Banach) tensor space equipped with a norm $\|\cdot\|$ is obtained by the completion (w.r.t. $\|\cdot\|$) of the algebraic tensor space

$$\overline{V}^{\|\cdot\|} = \overline{V_1 \otimes \ldots \otimes V_d}^{\|\cdot\|}.$$

If the V_ν are Hilbert spaces with inner products $(\cdot,\cdot)_\nu$ and associated norms $\|\cdot\|_\nu$, a canonical inner product on V can be first defined for elementary tensors

$$(v^{(1)} \otimes \ldots \otimes v^{(d)}, w^{(1)} \otimes \ldots \otimes w^{(d)}) = (v^{(1)}, w^{(1)}) \ldots (v^{(d)}, w^{(d)})$$

and then extended by linearity to the whole space V.

The associated norm $\|\cdot\|$ is called the canonical norm.

Infinite dimensional tensor spaces

Example (L^p spaces)

Let $1 \leq p < \infty$. If $V_\nu = L^p_{\mu_\nu}(\mathcal{X}_\nu)$, then

$$L^p_{\mu_1}(\mathcal{X}_1) \otimes \ldots \otimes L^p_{\mu_d}(\mathcal{X}_d) \subset L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$$

with $\mu = \mu_1 \otimes \ldots \otimes \mu_d$, and

$$\overline{L^p_{\mu_1}(\mathcal{X}_1) \otimes \ldots \otimes L^p_{\mu_d}(\mathcal{X}_d)}^{\|\cdot\|} = L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$$

where $\|\cdot\|$ is the natural norm on $L^p_\mu(\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$.

Example (Bochner spaces)

Let $\mathcal{X}$ be equipped with a finite measure μ, and let W be a Hilbert (or Banach) space. For $1 \leq p < \infty$, the Bochner space $L^p_\mu(\mathcal{X}; W)$ is the set of Bochner-measurable functions $u : \mathcal{X} \to W$ with bounded norm $\|u\|_p = (\int_{\mathcal{X}} \|u(x)\|^p_W \mu(dx))^{1/p}$, and

$$L^p_\mu(\mathcal{X}; W) = \overline{W \otimes L^p_\mu(\mathcal{X})}^{\|\cdot\|_p}.$$

Example (Sobolev spaces)

The Sobolev space $H^k(\mathcal{X})$ of functions defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$, equipped with the norm

$$\|u\|^2_{H^k} = \sum_{|\alpha|_1 \leq k} \|D^\alpha u\|^2_{L^2},$$

is a Hilbert tensor space

$$H^k(\mathcal{X}) = \overline{H^k(\mathcal{X}_1) \otimes \ldots \otimes H^k(\mathcal{X}_d)}^{\|\cdot\|_{H^k}}.$$

The Sobolev space $H^k_{mix}(\mathcal{X})$ equipped with the norm

$$\|u\|^2_{H^k_{mix}} = \sum_{|\alpha|_\infty \leq k} \|D^\alpha u\|^2_{L^2},$$

is a different tensor Hilbert space

$$H^k_{mix}(\mathcal{X}) = \overline{H^k(\mathcal{X}_1) \otimes \ldots \otimes H^k(\mathcal{X}_d)}^{\|\cdot\|_{H^k_{mix}}}.$$

$\|u\|_{H^k_{mix}}$ is the canonical tensor norm on $H^k(\mathcal{X}_1) \otimes \ldots \otimes H^k(\mathcal{X}_d)$.

Tensor product basis

If $\{\phi_i^{(\nu)}\}_{i \in I_\nu}$ is a basis of V_ν, then a basis of $V = V_1 \otimes \ldots \otimes V_d$ is given by

$$\left\{ \phi_i = \phi_{i_1}^{(1)} \otimes \ldots \otimes \phi_{i_d}^{(d)} : i \in I = I_1 \times \ldots \times I_d \right\}.$$

A tensor $v \in V$ admits a decomposition

$$v = \sum_{i \in I} a_i \phi_i = \sum_{i_1 \in I_1} \ldots \sum_{i_d \in I_d} a_{i_1, \ldots, i_d} \phi_{i_1}^{(1)} \otimes \ldots \otimes \phi_{i_d}^{(d)},$$

and v can be identified with the set of its coefficients

$$a \in \mathbb{R}^I.$$

Hilbert tensor spaces

If the $\{\phi_i^{(\nu)}\}_{i \in I_\nu}$ are orthonormal bases of spaces V_ν, then $\{\phi_i\}_{i\in I}$ is an orthonormal basis of the Hilbert tensor space $\overline{V}^{\|\cdot\|}$ equipped with the canonical norm. A tensor

$$v = \sum_{i \in I} a_i \phi_i$$

is such that

$$\|v\|^2 = \sum_{i \in I} a_i^2 := \|a\|^2.$$

Therefore, the map

$$a \mapsto \sum_{i \in I} a_i \phi_i$$

defines a linear isometry from $\ell_2(I)$ to V for finite dimensional spaces, and between $\ell_2(I)$ and $\overline{V}^{\|\cdot\|}$ for infinite dimensional spaces.

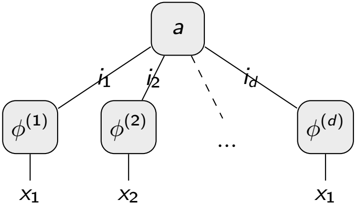
Tensor product feature map

If V is a space of functions defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$, we introduce the feature map $\phi^{(\nu)}(x_\nu) = (\phi^{(\nu)}_{i_\nu}(x_\nu))_{i_\nu \in I_\nu} \in \mathbb{R}^{I_\nu}$ and the tensor product feature map $\Phi : \mathcal{X} \to \mathbb{R}^I$ such that

$$\Phi(x) = \phi^{(1)}(x_1) \otimes \ldots \otimes \phi^{(d)}(x_d) \in \mathbb{R}^I$$

and a tensor v in V admits the representation

$$v(x) = (a, \Phi(x)) =$$



Outline

Rank of order-two tensors

The rank of an order-two tensor $u \in V \otimes W$, denoted $\text{rank}(u)$, is the minimal integer r such that

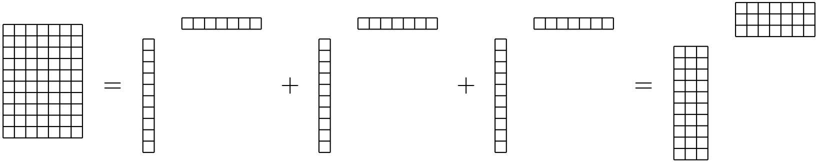
$$u = \sum_{k=1}^{r} v_k \otimes w_k$$

for some $v_k \in V$ and $w_k \in W$.

A tensor $u \in \mathbb{R}^n \otimes \mathbb{R}^m$ is identified with a matrix $u \in \mathbb{R}^{n \times m}$. The rank of u coincides with the matrix rank, which is the minimal integer r such that

$$u = \sum_{k=1}^{r} v_k w_k^T = VW^T,$$

where $V = (v_1, \ldots, v_r) \in \mathbb{R}^{n \times r}$ and $W = (w_1, \ldots, w_r) \in \mathbb{R}^{m \times r}$.



Singular value decomposition of order-two tensors

When V and W are Hilbert spaces (possibly infinite-dimensional), an algebraic tensor $u \in V \otimes W$ admits a singular value decomposition

$$u = \sum_{k \geq 1} \sigma_k v_k \otimes w_k,$$

where v_k and w_k are orthonormal vectors (singular vectors) and $\sigma_k \in \mathbb{R}^+$ are the singular values.

The rank of u is finite and coincides with the number of non-zero singular values,

$$\text{rank}(u) = \#\{k : \sigma_k \neq 0\}.$$

Example (Singular value decomposition of matrices)

For $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$, u is identified with a matrix in $\mathbb{R}^{n \times m}$ and

$$u = \sum_{k=1}^{\text{rank}(u)} \sigma_k v_k w_k^T = \mathsf{V}\mathsf{S}\mathsf{W}^T$$

with orthogonal matrices V and W, and a diagonal matrix S.

Singular value decomposition of order-two tensors

An algebraic tensor $u \in V \otimes W$ can be identified with a linear operator from W to V with rank equal to $\mathrm{rank}(u)$.

For infinite dimensional Hilbert spaces, the closure $\overline{V \otimes W}^{\|\cdot\|_\vee}$ of $V \otimes W$ with respect to the injective norm (corresponding to the operator norm or spectral norm) coincides with the space of compact operators.

A tensor $u \in \overline{V \otimes W}^{\|\cdot\|_\vee}$ still admits a singular value decomposition

$$u = \sum_{k \geq 1} \sigma_k v_k \otimes w_k.$$

and the rank (number of non-zero singular values) is possibly infinite.

Singular value decomposition of order-two tensors

Example (Proper Orthogonal Decomposition)

For $\Omega \times I$ a space-time domain and V a Hilbert space of functions defined on Ω, a function $u \in L^2(I; V)$ admits a singular value decomposition

$$u(t) = \sum_{k=1}^{\infty} \sigma_k v_k w_k(t)$$

which is known as the Proper Orthogonal Decomposition (POD).

Example (Karhunen-Loeve decomposition)

For a probability space (Ω, μ), an element $u \in L^2_\mu(\Omega; V)$ is a second-order V-valued random variable. If u is zero-mean, the singular value decomposition of u is known as the Karhunen-Loeve decomposition

$$u(\omega) = \sum_{k=1}^{\infty} \sigma_k v_k w_k(\omega)$$

where $w_k : \Omega \to \mathbb{R}$ are uncorrelated (orthogonal) random variables.

Low-rank format for order-two tensors

The set of tensors in $V \otimes W$ with rank bounded by r, denoted

$$\mathcal{R}_r = \{v : \operatorname{rank}(v) \leq r\},$$

is not a linear space nor a convex set. However, it has many favorable properties for a numerical use.

- The application $v \mapsto \operatorname{rank}(v)$ is lower semi-continuous, and therefore the set $\mathcal{R}_r$ is closed, which makes best approximation problems in $\mathcal{R}_r$ well posed.
- $\mathcal{R}_r$ is the union of smooth manifolds of tensors with fixed rank.

Canonical rank of higher-order tensors

For tensors $u \in V_1 \otimes \ldots \otimes V_d$ with $d \geq 3$, there are different notions of rank.

The canonical rank, which is the natural extension of the notion of rank for order-two tensors, is the minimal integer r such that

$$u(x_1, \ldots, x_d) = \sum_{k=1}^{r} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d),$$

for some vectors $v_k^{(\nu)} \in V_\nu$.

Example

- A monomial $x^i = x_1^{i_1} \ldots x_d^{i_d}$ has rank 1.
- A polynomial $\sum_{i \in \Lambda} a_i x^i$ has rank $\#\Lambda$.
- A Gaussian function $\exp(-\alpha \|x - a\|_2^2) = \prod_{i=1}^{d} \exp(-\alpha(x_i - a_i)^2)$ has rank 1.
- The function $\frac{1}{\|x\|_2}$ has infinite rank.

Canonical format

The subset of tensors in $V = V_1 \otimes \ldots \otimes V_d$ with canonical rank bounded by r is denoted

$$\mathcal{R}_r = \{v \in V : \operatorname{rank}(v) \leq r\}.$$

A tensor in $\mathcal{R}_r$ has a representation

$$v(x_1, \ldots, x_d) = \sum_{k=1}^{r} v_k^{(1)}(x_1) \ldots v_k^{(d)}(x_d)$$

The storage complexity of tensors in $\mathcal{R}_r$ is

$$\operatorname{storage}(\mathcal{R}_r) = r \sum_{\nu=1}^{d} \dim(V_\nu) = O(rdn)$$

for $\dim(V_\nu) = O(n)$.

Canonical format

For $d \geq 3$, the set $\mathcal{R}_r$ looses many of the favorable properties of the case $d = 2$.

- Determining the rank of a given tensor is a NP-hard problem.
- The set $\mathcal{R}_r$ is not an algebraic variety.
- No notion of singular value decomposition.
- The application $v \mapsto \mathrm{rank}(v)$ is not lower semi-continuous and therefore, $\mathcal{R}_r$ is not closed.

Example

Consider the order-3 tensor

$$v = a \otimes a \otimes b + a \otimes b \otimes a + b \otimes a \otimes a$$

where a and b are linearly independent vectors in $\mathbb{R}^m$. The rank of v is 3. The sequence of rank-2 tensors

$$v_n = n(a + \frac{1}{n}b) \otimes (a + \frac{1}{n}b) \otimes (a + \frac{1}{n}b) - na \otimes a \otimes a$$

converges to v as $n \to \infty$.

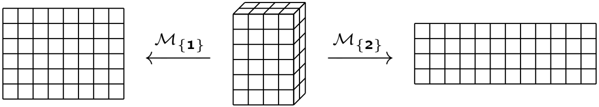
- The consequence is that for most problems involving approximation in canonical format $\mathcal{R}_r$, there is no robust method when $d > 2$.

α-rank

For a non-empty subset α of $D = \{1, \ldots, d\}$, a tensor $u \in V = V_1 \otimes \ldots \otimes V_d$ can be identified with an order-two tensor

$$\mathcal{M}_\alpha(u) \in V_\alpha \otimes V_{\alpha^c},$$

where $V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$, and $\alpha^c = D \setminus \alpha$. The operator $\mathcal{M}_\alpha = V \to V_\alpha \otimes V_{\alpha^c}$ is called the matricisation (or unfolding) operator.



The α-rank of u, denoted $\mathrm{rank}_\alpha(u)$, is the rank of the order-two tensor $\mathcal{M}_\alpha(u)$,

$$\mathrm{rank}_\alpha(u) = \mathrm{rank}(\mathcal{M}_\alpha(u)),$$

which is the minimal integer r_α such that

$$\mathcal{M}_\alpha(u) = \sum_{k=1}^{r_\alpha} v_k^\alpha \otimes w_k^{\alpha^c}$$

for some $v_k^\alpha \in V_\alpha$ and $w_k^{\alpha^c} \in V_{\alpha^c}$. We note that $\mathrm{rank}_\alpha(u) = \mathrm{rank}_{\alpha^c}(u)$.

A multivariate function $u(x_1, \dots, x_d)$ with $\mathrm{rank}_\alpha(u) \le r_\alpha$ is such that

$$u(x) = \sum_{k=1}^{r_\alpha} v_k^\alpha(x_\alpha) w_k^{\alpha^c}(x_{\alpha^c})$$

for some functions $v_k^\alpha(x_\alpha)$ and $w_k^{\alpha^c}(x_{\alpha^c})$ of groups of variables

$$x_\alpha = \{x_\nu\}_{\nu \in \alpha} \quad \text{and} \quad x_{\alpha^c} = \{x_\nu\}_{\nu \in \alpha^c}.$$

α-rank

Example

- $u(x) = u^1(x_1)\dots u^d(x_d)$ can be written $u(x) = u^\alpha(x_\alpha)u^{\alpha^c}(x_{\alpha^c})$, with $u^\alpha(x_\alpha) = \prod_{\nu\in\alpha} u^\nu(x_\nu)$. Therefore, for any α, $\text{rank}_\alpha(u) = 1$.

- $u(x) = \sum_{k=1}^r u_k^1(x_1)\dots u_k^d(x_d)$ can be written $\sum_{k=1}^r u_k^\alpha(x_\alpha)u_k^{\alpha^c}(x_{\alpha^c})$ with $u_k^\alpha(x_\alpha) = \prod_{\nu\in\alpha} u_k^\nu(x_\nu)$. Therefore, for any α, $\text{rank}_\alpha(u) \le r$, with equality if the functions $\{u_k^\alpha(x_\alpha)\}$ and the functions $\{u_k^{\alpha^c}(x_{\alpha^c})\}$ are linearity independent.

  We deduce the following relation between α-ranks and canonical rank:

  $$\text{rank}_\alpha(u) \le \text{rank}(u), \quad \text{for any } \alpha.$$

- $u(x) = u^1(x_1) + \dots + u^d(x_d)$ can be written $u(x) = u^\alpha(x_\alpha) + u^{\alpha^c}(x_{\alpha^c})$, with $u^\alpha(x_\alpha) = \sum_{\nu\in\alpha} u^\nu(x_\nu)$. Therefore, $\text{rank}_\alpha(u) \le 2$.

- $u(x) = \prod_{\alpha\in T} u^\alpha(x_\alpha)$ with T a collection of disjoint subsets, is such that $\text{rank}_\alpha(u) = 1$ for all $\alpha \in T$, and $\text{rank}_\gamma(u) \le \prod_{\alpha\in T, \alpha\cap\gamma\neq\emptyset} \text{rank}_{\gamma\cap\alpha}(u^\alpha)$ for all γ.

α-ranks and minimal subspaces

For a subset α of $D = \{1, \ldots, d\}$, the minimal subspace

$$U_\alpha^{min}(u)$$

of a tensor $u \in V_1 \otimes \ldots \otimes V_d$ is defined as the smallest subspace

$$U_\alpha \subset V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$$

such that

$$\mathcal{M}_\alpha(u) \in U_\alpha \otimes V_{\alpha^c}.$$

The α-rank of u is the dimension of the minimal subspace $U_\alpha^{min}(u)$,

$$\text{rank}_\alpha(u) = \dim(U_\alpha^{min}(u)).$$

If u admits the representation

$$u(x) = \sum_{k=1}^{\text{rank}_\alpha(v)} v_k^\alpha(x_\alpha) v^{\alpha^c}(x_{\alpha^c})$$

then $U_\alpha^{min}(u) = span\{v_k^\alpha : 1 \leq k \leq \text{rank}_\alpha(u)\}$.

For any partition $\{\alpha_1, \dots, \alpha_m\}$ of D, an algebraic tensor u is such that

$$u \in U_{\alpha_1}^{min}(u) \otimes \dots \otimes U_{\alpha_m}^{min}(u)$$

Moreover, for any $\alpha \subset D$ and any partition $\{\beta_1, \dots, \beta_s\}$ of α, it holds

$$U_{\alpha}^{min}(u) \subset U_{\beta_1}^{min}(u) \otimes \dots \otimes U_{\beta_s}^{min}(u)$$

that implies

$$\mathrm{rank}_{\alpha}(u) \leq \prod_{k=1}^{s} \mathrm{rank}_{\beta_k}(u)$$

Also, for any $p \in \{1, ..., s\}$

$$\mathrm{rank}_{\beta_p}(u) \leq \mathrm{rank}_{\alpha}(u) \prod_{\substack{k=1 \\ k \neq p}}^{s} \mathrm{rank}_{\beta_k}(u)$$

Example

The function

$$u(x_1, x_2, x_3) = \cos(x_1 + x_2) + x_1(x_2 + 2x_3) = \cos(x_1)\cos(x_2) - \sin(x_1)\sin(x_2) + x_1x_2 + 2x_1x_3$$

has for minimal subspaces and ranks

- $U_1^{min}(u) = span\{\cos(x_1), \sin(x_1), x_1\}, \quad r_1 = 3$
- $U_2^{min}(u) = span\{\cos(x_2), \sin(x_2), x_2\}, \quad r_2 = 3$
- $U_3^{min}(u) = span\{1, x_3\}, \quad r_3 = 2$
- $U_{1,2}^{min}(u) = span\{\cos(x_1 + x_2), x_1x_2, x_1\}, \quad r_{1,2} = 3$
- $U_{2,3}^{min}(u) = span\{\cos(x_2), \sin(x_2), x_2 + 2x_3\}, \quad r_{2,3} = 3$
- $U_{1,3}^{min}(u) = span\{\cos(x_1), \sin(x_1), x_1, x_1x_3\}, \quad r_{1,3} = 4$

In particular, we can check that

$$U_{1,3}^{min}(u) \subset U_1^{min}(u) \otimes U_3^{min}(u) = span\{\cos(x_1), \sin(x_1), x_1, \cos(x_1)x_3, \sin(x_1)x_3, x_1x_3\}$$

$$r_{1,3} \leq r_1 r_3, \quad r_1 \leq r_{1,3} r_3, \quad r_3 \leq r_{1,3} r_1$$

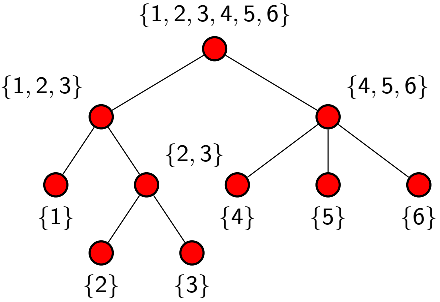
Outline

Tree-based tensor format

Tree-based (Hierarchical) tensor formats [Hackbusch-Kuhn'09] are subsets of tensors

$$\mathcal{T}_r^T = \{v \in V : \mathrm{rank}_\alpha(v) \le r_\alpha, \alpha \in T\}$$

where $r = (r_\alpha)_{\alpha \in T}$ and where T is a dimension partition tree T over $D = \{1, \dots, d\}$, with root D and leaves $\mathcal{L}(T) = \{\{\nu\} : 1 \le \nu \le d\}$. All nodes in T are non empty subsets of D. The set of children of $\alpha \in T$ is either empty (for a leaf node) or is a nontrivial partition of α (for an interior node).



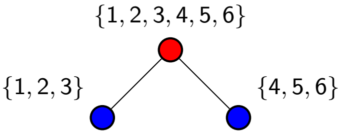
The tree-based rank of a tensor v is the tuple $\mathrm{rank}_T(v) = (\mathrm{rank}_\alpha(v))_{\alpha \in T}$.

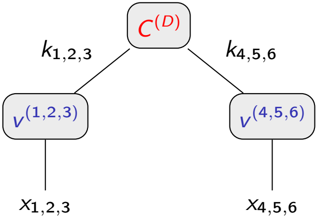
By convention, $\mathrm{rank}_D(v) = 1$.

Elements of $\mathcal{T}_r^T$ admit an explicit representation. Let $v \in \mathcal{T}_r^T$ with T-rank $r = (r_\alpha)_{\alpha \in T}$. At the first level, v admits the representation

$$v(x) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \dots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(D)}_{k_{\beta_1},\dots,k_{\beta_s}} v^{(\beta_1)}_{k_{\beta_1}}(x_{\beta_1}) \dots v^{(\beta_s)}_{k_{\beta_s}}(x_{\beta_s})$$

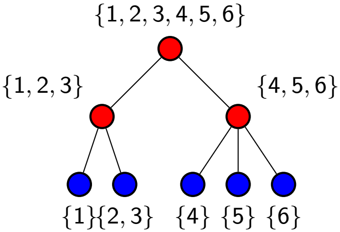
where $\{\beta_1, \dots, \beta_s\} = S(D)$ are the children of the root node D, and $\{v^{(\beta)}_{k_\beta}\}_{1 \le k_\beta \le r_\beta}$ form a basis of the minimal subspace $U^{min}_\beta(v)$.

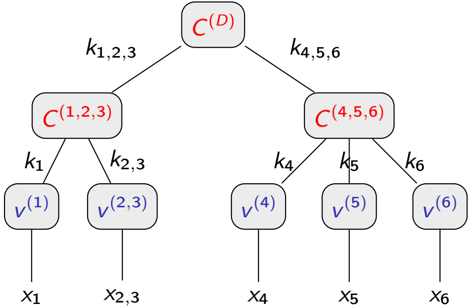




Then, for an interior node α of the tree, with children $S(\alpha) = \{\beta_1, \dots, \beta_s\}$, the functions (or tensors) $v^{(\alpha)}_{k_\alpha}$ admit the representation

$$v^{(\alpha)}_{k_\alpha}(x_\alpha) = \sum_{k_{\beta_1}=1}^{r_{\beta_1}} \dots \sum_{k_{\beta_s}=1}^{r_{\beta_s}} C^{(\alpha)}_{k_\alpha, k_{\beta_1}, \dots, k_{\beta_s}} v^{(\beta_1)}_{k_{\beta_1}}(x_{\beta_1}) \dots v^{(\beta_s)}_{k_{\beta_s}}(x_{\beta_s}).$$



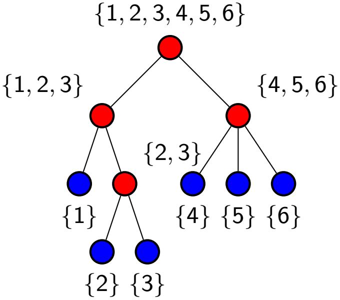


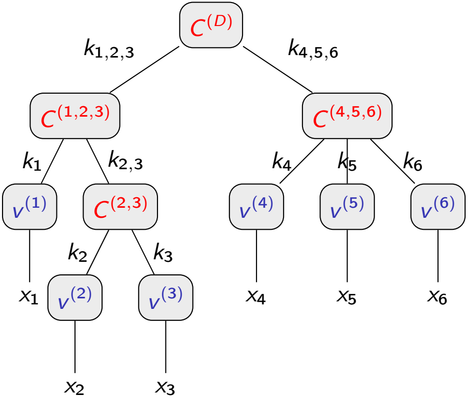
Tree-based tensor format as a tree tensor network

Finally, the tensor v admits the representation

$$v(x) = \sum_{\substack{1 \le k_\beta \le r_\beta \\ \beta \in T}} \prod_{\alpha \in T \setminus \mathcal{L}(T)} C^{(\alpha)}_{(k_\beta)_{\beta \in S(\alpha)}, k_\alpha} \prod_{\nu \in \mathcal{L}(T)} v^{(\nu)}_{k_\nu}(x_\nu)$$

where the parameters C^α and $v^{(\nu)}$ form a tree tensor network.



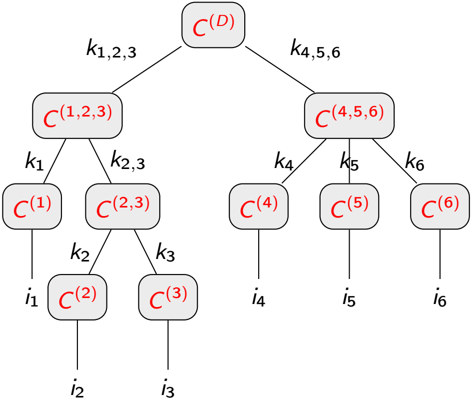


Given bases $\{\phi^{\alpha}_{i_\alpha}(x_\alpha)\}_{i_\alpha \in I^\alpha}$ of functions for the spaces V_α for $\alpha \in \mathcal{L}(T)$,

$$v(x) = \sum_{i_1 \in I^1} \dots \sum_{i_d \in I^d} a(i_1, \dots, i_d)\phi_{i_1}(x_1)\dots\phi_{i_d}(x_d)$$

with $a(i_1, \dots, i_d) = \sum_{\substack{1 \le k_\beta \le r_\beta \\ \beta \in T}} \prod_{\alpha \in T \setminus \mathcal{L}(T)} C^{(\alpha)}_{(k_\beta)_{\beta \in S(\alpha)}, k_\alpha} \prod_{\alpha \in \mathcal{L}(T)} C^{(\alpha)}_{i_\alpha, k_\alpha}$ or using tensor diagram notations

$a(i_1, \dots, i_d) =$



Representation complexity

The representation complexity for the representation of a tensor in $\mathcal{T}_r^T(V)$ is

$$C(T, r) = \sum_{\alpha \in T \setminus \mathcal{L}(T)} r_\alpha \prod_{\beta \in S(\alpha)} r_\beta + \sum_{\nu \in \mathcal{L}(T)} \# I^\alpha r_\alpha .$$

If $r_\alpha = O(R)$ and $\# I^\alpha = O(N)$,

$$C(T, r) = O(dNR + (\# T - d - 1) R^{s+1} + R^s),$$

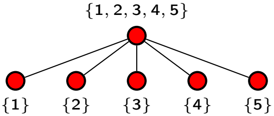
where $s = \max_{\alpha \in T \setminus \mathcal{L}(T)} \# S(\alpha)$ is the arity of the tree.

Since $\# T \leq 2d + 1$,

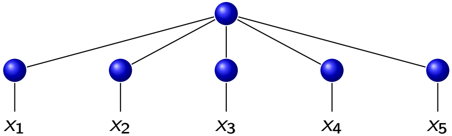
$$C(T, r) = O(dNR + dR^{s+1} + R^s)$$

Tucker format

The Tucker format [Hitchcock'27] corresponds to a trivial tree with one level, arity $s = d$, $\#T = d + 1$,



The representation of a tensor u in $\mathcal{T}_r^T$ is

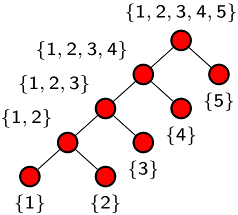


The representation complexity

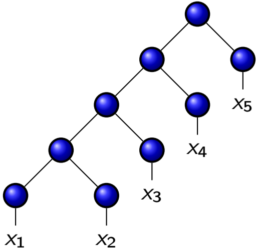
$$C(T, r) = O(dNR + R^d)$$

Tensor train Tucker format

The tensor train Tucker format corresponds to a linear binary tree



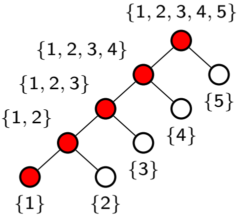
The representation of a tensor u in $\mathcal{T}_r^T$ is



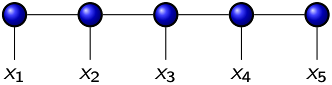
The representation complexity $C(T, r) = O(dNR + (d-2)R^3 + R^2)$.

The tensor train format [Oseledets-Tyrtyshnikov'09] was discovered independently in quantum physics [Baxter'68 , Affleck'87] and coined Matrix Product State (MPS).
It corresponds to a degenerate tree-based format where T is a subset of a linear tree

$$T = \{\{1\}, \{1,2\}, \{1,2,3\}, \dots, \{1,\dots,d\}\}$$



The representation of a tensor u in $\mathcal{T}_r^T$ is



or explicitly

$$u(x_1,\dots,x_d) = \sum_{k_1=1}^{r_1} \dots \sum_{k_{d-1}=1}^{r_{1,\dots,d-1}} v^{(1)}_{k_1}(x_1) v^{(2)}_{k_1,k_2}(x_2) \dots v^{(d-1)}_{k_{d-2},k_{d-1}}(x_{d-1}) v^{(d)}_{k_{d-1}}(x_d)$$

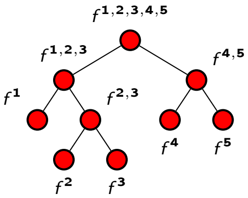
The complexity is $C(T,r) = O(dNR^2)$.

Tree tensor networks as a compositional function network

By identifying a tensor $C^{(\alpha)} \in \mathbb{R}^{n_1 \times \ldots \times n_s \times r_\alpha}$ with a $\mathbb{R}^{r_\alpha}$-valued multilinear function

$$f^{(\alpha)} : \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_s} \to \mathbb{R}^{r_\alpha},$$

a function v in $\mathcal{T}_r^T$ admits a representation as a tree-structured composition of multilinear functions $\{f^{(\alpha)}\}_{\alpha \in T}$.



$$v(x) = f^D(f^{1,2,3}(f^1(\Phi^1(x_1)), f^{2,3}(f^2(\Phi^2(x_2)), f^3(\Phi^3(x_3))), f^{4,5}(f^4(\Phi^4(x_4)), f^5(\Phi^5(x_5))))$$

where $\Phi^\nu(x_\nu) = (\phi^\nu_{i_\nu}(x_\nu))_{i_\nu \in I^\nu} \in \mathbb{R}^{\# I^\nu}$.

Tree tensor networks as feed-forward neural networks

It corresponds to a sum-product feed forward neural network with a sparse architecture (given by T), a number of hidden layers equal to $depth(T) + 1$ (including a featuring layer), and width at level ℓ related to the α-ranks of the nodes α of level ℓ.

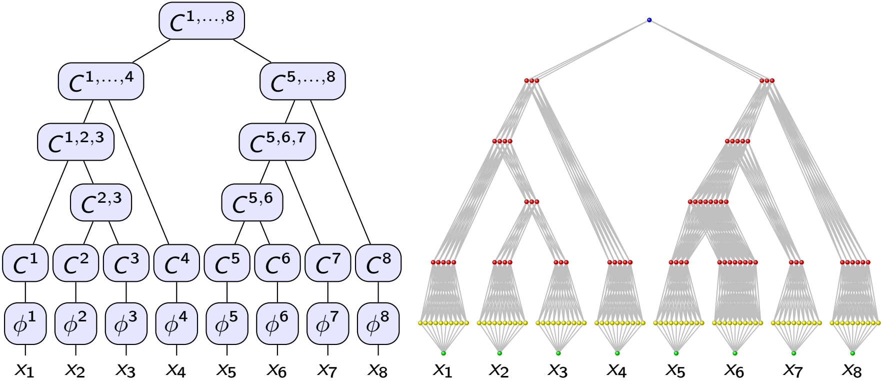


Figure: Tree tensor network and corresponding feed-forward sum-product neural network with 10 features per variable x_ν (right)

Properties of tree-based tensor formats

Many favorable properties inherited from the matrix case.

- Complexity is linear in d and polynomial in the rank for storage, evaluation, differentiation, integration...
- Not so nonlinear approximation tool. A tensor u in tree-based format admits a multilinear parametrization with parameters $(C_\alpha)_{\alpha \in T}$ forming a tree tensor network, i.e.

  $$u = R((C_\alpha)_{\alpha \in T})$$

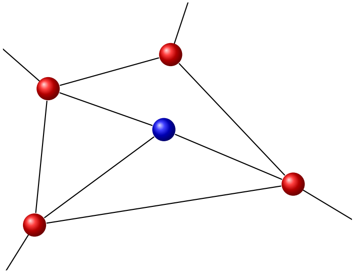
  with R a multilinear map.
- Topological properties ensure the well-posedness of optimization problems and existence of stable algorithms
- Geometrical properties can be exploited for optimization and dynamical approximation.
- Possible extensions of singular value decomposition for u in a Hilbert tensor space V, and a way to obtain approximations u_r in $\mathcal{T}_r^T(V)$ such that

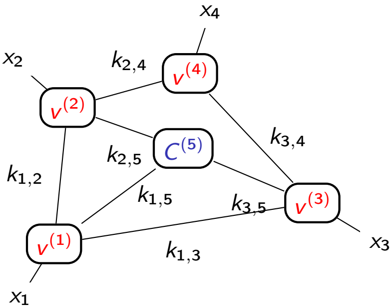
  $$\|u - u_r\| \leq C_d \inf_{v \in \mathcal{T}_r^T(V)} \|u - v\|$$

  with $C_d \sim \sqrt{d}$.

General tensor networks

More general tensor networks are associated with graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with nodes (vertices) $\mathcal{N}$ and edges $\mathcal{E}$, d of the nodes being associated with variables x_ν, $1 \leq \nu \leq d$





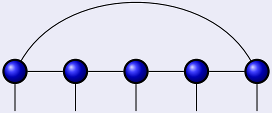
They have a multilinear parametrization of the form

$$v(x_1, \ldots, x_d) = \sum_{\substack{1 \leq k_e \leq r_e \\ e \in \mathcal{E}}} \prod_{\nu=1}^{d} v^{(\nu)}(x_\nu, (k_e)_{e \in E_\nu}) \prod_{\nu=d+1}^{N} C^{(\nu)}((k_e)_{e \in E_\nu})$$

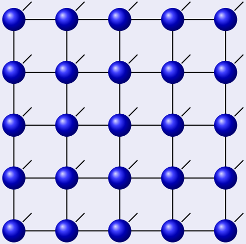
Tree tensor networks is a particular case where $\mathcal{G}$ is a tree.

Examples of tensor networks

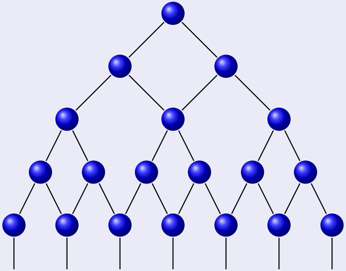
Tensor ring (MPS with periodic boundary conditions)



PEPS



MERA



General tensor networks

When the graph contains cycles,

- integers r_e (bond dimensions) may not have an interpretation as α-ranks,
- no notion of singular value decomposition,
- loss of nice geometrical and topological properties,
- computational complexity increases,
- but yet powerful for some high-dimensional applications.

Outline

Tensorization of vectors

A vector $v \in \mathbb{R}^N$ with $N = b^L$ can be identified with a tensor of order L

$$\boldsymbol{v} \in \mathbb{R}^b \otimes \ldots \otimes \mathbb{R}^b = (\mathbb{R}^b)^{\otimes L}$$

such that for $i \in \{0, \ldots, N-1\}$

$$v(i) = \boldsymbol{v}(i_1, \ldots, i_L)$$

where $(i_1, \ldots, i_L) \in \{0, \ldots, b-1\}$ are the integers of the representation of i in base b

$$i = \sum_{k=1}^{d} i_k b^{L-k} = [i_1, \ldots, i_L]_b.$$

The map which associates to v its tensorization $\boldsymbol{v}$ is a linear isometry from $\ell_2(\mathbb{R}^N)$ to $\ell_2(\mathbb{R}^b)^{\otimes L}$.

Some matrix-vector operations can be efficiently implemented using tensor algebra, such as the Hadamard transform

$$H_L v \equiv (H_1 \otimes \ldots \otimes H_1)\boldsymbol{v}$$

Tensorization of tensors

A tensor $v \in \mathbb{R}^N \otimes \ldots \otimes \mathbb{R}^N = (\mathbb{R}^N)^{\otimes d}$ with $N = b^L$ can be identified with a tensor of order dL

$$\boldsymbol{v} \in (\mathbb{R}^b)^{\otimes dL}$$

with

$$v(i_1, \ldots, i_d) = \boldsymbol{v}(i_1^1, \ldots, i_1^L, \ldots, i_d^1, \ldots, i_d^L)$$

where

$$i_\nu = [i_\nu^1 \ldots i_\nu^{L_\nu}]_b$$

Other orderings of variables can be considered, such as

$$v(i_1, \ldots, i_d) = \boldsymbol{v}(i_1^1, \ldots, i_d^1, \ldots, i_1^L, \ldots, i_d^L)$$

Tensors with different dimensions can be considered, i.e.

$$v \in \mathbb{R}^{N_1} \otimes \ldots \otimes \mathbb{R}^{N_d}, \quad N_\nu = b_\nu^{L_\nu}$$

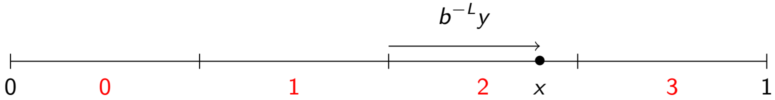
is identified with a tensor of order $\sum_{\nu=1}^d L_\nu$.

Tensorization of univariate functions

Consider a function $f \in \mathbb{R}^{[0,1)}$ defined on the interval $[0,1)$.

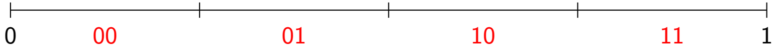
- For $b, L \in \mathbb{N}$, we subdivide uniformly the interval $[0,1)$ into b^L intervals. Any $x \in [0,1)$ can be written

$$x = b^{-L}(i + y), \quad i \in \{0, \dots, b^L - 1\}, \quad y \in [0,1).$$



- The integer i admits a representation in base b

$$i = \sum_{k=1}^{L} i_k b^{L-k} = [i_1 \dots i_L]_b, \quad i_k \in \{0, \dots, b-1\}$$

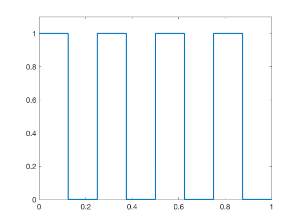


- f is thus identified with a multivariate function (tensor of order $L+1$)

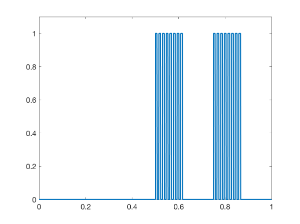
$$\boldsymbol{f} \in (\mathbb{R}^b)^{\otimes L} \otimes \mathbb{R}^{[0,1)} \quad \text{such that} \quad f(x) = \boldsymbol{f}(i_1, \dots, i_L, y)$$

Tensorization of univariate functions

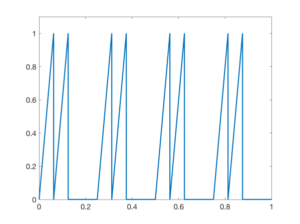
Examples of elementary tensors $f(x) = v^1(i_1)...v^L(i_L)v^{L+1}(y)$ $(b = 2)$



(a) $\delta_{\mathbf{0}}(i_{\mathbf{3}})$



(b) $\delta_{\mathbf{1}}(i_{\mathbf{1}})\delta_{\mathbf{0}}(i_{\mathbf{3}})\delta_{\mathbf{0}}(i_{\mathbf{7}})$



(c) $\delta_{\mathbf{0}}(i_{\mathbf{1}})y$ $(L = 4)$

Ranks of polynomials and splines

Polynomials

Consider a polynomial $q(x)$ of degree p. For any $\alpha \subset \{1, \dots, L\}$,

$$q(x) = q(b^{-L}(\sum_{k=1}^{L} i_k b^{L-k} + y)) = q(g(i_\alpha) + \tilde{g}(i_{\alpha^c})) = \sum_{j=0}^{p} g(i_\alpha)^j h_j(i_{\alpha^c})$$

so that $\mathrm{rank}_\alpha(\boldsymbol{q}) \leq p + 1$.

Trigonometric polynomials

The tensorization of function $\cos(\omega x + \varphi)$ at resolution L has all ranks equal to 2.

Then a trigonometric polynomial $q(x)$ of degree p is such that for any $\alpha \subset \{1, \dots, L\}$,

$$\mathrm{rank}_\alpha(\boldsymbol{q}) \leq 2p + 1.$$

Ranks of polynomials and splines

Splines

A spline φ_N of degree p over N b-adic intervals forming a partition of $[0,1)$ is such that

$$\mathrm{rank}_{\{1,\ldots,\nu\}}(\varphi_N) \leq \begin{cases} p+N, & 1 \leq \nu < \ell. \\ p+1, & \ell \leq \nu \leq L. \end{cases}$$

where $b^{-\ell}$ is the minimal length of intervals.

Tensorization of multivariate functions

A function $f(x_1, \ldots, x_d)$ defined on $[0,1)^d$ can be similarly identified with a tensor of order $(L+1)d$

$$\boldsymbol{f} \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \ldots, x_d) = \boldsymbol{f}(i_1^1, \ldots, i_d^1, \ldots, i_1^L, \ldots, i_d^L, y_1, \ldots, y_d)$$

$$\text{where} \quad x_\nu = b^{-L}(\sum_{k=1}^{L} i_\nu^k b^{L-k} + y_\nu) = b^{-L}([i_\nu^1 \ldots i_\nu^L]_b + y_\nu)$$

The map $T_{b,d}$ which associates to a function f its tensorization $\boldsymbol{f}$ is a linear isometry from $L^p([0,1)^d)$ to $L^p(\{0,\dots,b-1\}^{Ld} \times [0,1)^d)$ for any $0 < p \leq \infty$.

References I

W. Hackbusch.
Tensor Spaces and Numerical Tensor Calculus, volume 56.
Springer Nature, 2019.

T. G. Kolda and B. W. Bader.
Tensor decompositions and applications.
SIAM Review, 51(3):455–500, September 2009.

L.-H. Lim.
Tensors in computations.
arXiv e-prints, page arXiv:2106.08090, June 2021.

A. Nouy.
Low-rank methods for high-dimensional approximation and model order reduction.
In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), Model Reduction and Approximation: Theory and Algorithms. SIAM, Philadelphia, PA, 2016.

R. Orus.
A practical introduction to tensor networks: Matrix product states and projected entangled pair states.
Annals of Physics, 349:117 – 158, 2014.

CEMRACS,
July 19-23, 2021

Approximation and learning with tensor networks

Part II: Approximation theory of tree tensor networks

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Outline

Outline

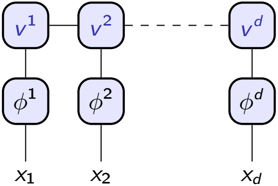
Approximation tools based on tree tensor networks

For the approximation of a target function $u(x_1, \dots, x_d)$, a first approach is to introduce subspaces $V^{\nu}_{N_\nu}$ of finite dimension (e.g. polynomials, splines, wavelets...) and consider tensor networks $f \in \mathcal{T}_r^T(V_N)$ with

$$V_N = V^1_{N_1} \otimes \dots \otimes V^d_{N_d}$$

e.g. with the tensor train format

$f(x_1, \dots, x_d) =$



with ϕ^ν a feature map associated with $V^\nu_{N_\nu}$.

Spaces $V^\nu_{N_\nu}$ have to be well chosen, e.g. polynomials for analytic functions, splines with a degree adapted to the regularity of the target function...

An approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$ is then defined by

$$\Phi_n = \{f \in \mathcal{T}_r^T(V_N) : N \in \mathbb{N}^d, r \in \mathbb{N}^T, compl(f) \leq n\}.$$

The dimensions N and the ranks r are free parameters, and $compl(\cdot)$ is some complexity measure.

Approximation tools based on tree tensor networks

An alternative is to rely on tensorization of functions. A d-variate function f is identified with a tensor

$$\boldsymbol{f} = T_{b,d}(f) \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \dots, x_d) = \boldsymbol{f}(i_1^1, \dots, i_d^1, \dots, i_1^L, \dots, i_d^L, y_1, \dots, y_d) \quad \text{with} \quad x_\nu = b^{-L}([i_\nu^1 \dots i_\nu^L]_b + y_\nu).$$

Then we consider functions whose tensorization at resolution L are in the tensor space

$$\boldsymbol{V}_L = (\mathbb{R}^b)^{\otimes Ld} \otimes S^{\otimes d}$$

with $S \subset \mathbb{R}^{[0,1)}$ some subspace of univariate functions.

If $S = \mathbb{P}_m$, $V_L = T_{b,d}^{-1}(\boldsymbol{V}_L)$ is identified with the space of multivariate splines of degree m over a uniform partition with b^{dL} elements, i.e.

$$V_L = V_{N_1}^1 \otimes \dots \otimes V_{N_d}^d$$

with $N_1 = \dots = N_d = b^L$ and $V_{N_\nu}^\nu$ a space of univariate splines of degree m over a uniform partition with $N_\nu = b^L$ intervals.

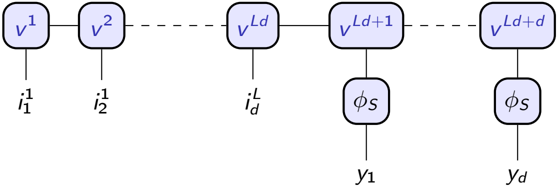
Note that different resolutions L_ν could be used to tensorize the different variables x_ν.

Approximation tools based on tree tensor networks

Then as an approximation tool, we consider functions f whose tensorization is a tensor network in $\mathcal{T}_r^{T_L}(\boldsymbol{V}_L)$, with T_L a dimension tree over $\{1, \dots, Ld + d\}$.

Using the tensor train format, the corresponding function $f(x_1, \dots, x_d)$ has the representation

$f(x_1, \dots, x_d) =$



with ϕ_S the feature map associated with S. This is similar to the quantized tensor train (QTT) format [Kazeev, Khoromskij, Oseledets, Schwab, ...]

Later on, we consider $S = \mathbb{P}_m$ and $\phi_S(y) = (1, y, ..., y^{m+1})$ or any other polynomial basis.

An approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$ is then defined by

$$\Phi_n = \{f \in \Phi_{L,T_L,r} : L \in \mathbb{N}_0, r \in \mathbb{N}^{T_L}, compl(f) \leq n\}$$

with $\Phi_{L,T_L,r}$ the functions whose tensorization at resolution L is in $\mathcal{T}_r^{T_L}(V_L)$.

The resolution L and ranks r are free parameters, and $compl(\cdot)$ is some complexity measure.

Complexity measures and corresponding approximation tools

The complexity $compl(f)$ of f is defined as the complexity of the associated tensor network $\boldsymbol{v} = \{v^\alpha\}_{\alpha \in T}$.

- Number of parameters (full tensors network)

$$compl_{\mathcal{F}}(f) = \sum_{\alpha} \text{number\_of\_entries}(v^\alpha)$$

- Number of non-zero parameters (sparse tensors network)

$$compl_{\mathcal{S}}(f) = \sum_{\alpha} \|v^\alpha\|_0$$

Complexity measures $compl_{\mathcal{F}}$ and $compl_{\mathcal{S}}$ yield two different approximation tools

$$\Phi_n^{\mathcal{F}} \quad \text{and} \quad \Phi_n^{\mathcal{S}}$$

such that

$$\Phi_n^{\mathcal{F}} \subset \Phi_n^{\mathcal{S}}$$

Approximation with tree tensor networks

Given a function f from a Banach space X, the best approximation error of f by an element of Φ_n is

$$E(f, \Phi_n)_X := \inf_{g \in \Phi_n} \|f - g\|_X$$

Fundamental questions are:

- does $E(f, \Phi_n)_X$ converge to 0 for any f ?
  (universality)
- does a best approximation exist ?
  (proximinality)
- how fast does it converge for functions from classical function classes ?
  (expressivity)
- what are the functions for which $E(f, \Phi_n)_X$ converges with some given rate ?
  (characterization of approximation classes)

Another fundamental problem (addressed later) is to provide algorithms to practically compute approximations using available information on the function (model equations, samples...)

Outline

Universality

First note that for any algebraic feature tensor space V, and any tree T,

$$\bigcup_r \mathcal{T}_r^T(V) = V.$$

so the question of universality of tree tensor networks boils down to conditions on the tensor feature spaces.

- Consider the first family of approximation tools with variable feature spaces V_N, $N \in \mathbb{N}^d$.

  If $\bigcup_N V_N$ is dense in X, then the tools are universal for functions in X.

  In particular, this is true for $X = L^p((0,1)^d)$, $p < \infty$, and for polynomial or splines spaces V_N.

- Consider the second family of approximation tools using tensorization.

  If $\bigcup_L V_L$ is dense in X, then the tools are universal for functions in X.

  In particular, this is true for $X = L^p((0,1)^d)$, $p < \infty$, assuming that S contains the function one.

Proximinality

For any tree T, any T-rank r, and any finite dimensional tensor space V of X, $\mathcal{T}_r^T(V)$ is a closed set in V.

Φ_n is a finite union of such sets, all contained in a single finite dimensional space V^*. Then Φ_n is a closed set of a finite dimensional space V^* and is therefore proximinal in X.

Expressivity

Different ways to analyse the expressivity of tree tensor networks

- Exploit known results on other approximation tools and estimate the complexity to encode these tools using tree tensor networks.
- Directly encode a function using tree tensor networks (with controlled errors)
- Analyse the convergence of bilinear approximations

$$u(x_\alpha, x_{\alpha^c}) \approx \sum_{k=1}^{r_\alpha} u_k^\alpha(x_\alpha) u_k^{\alpha^c}(x_{\alpha^c})$$

  or the approximability of partial evaluations $u(\cdot, x_{\alpha^c})$ by linear approximation spaces of dimension r_α

Approximation of functions from smoothness classes

We consider approximation tools based on tensorization and functions from classical smoothness classes:

- Sobolev and Besov functions
- Analytic functions
- Analytic functions with singularities

Approximation of functions from Besov spaces $B^\alpha_q(L^p)$

From results on spline approximation and their encoding with tensor networks, we obtain

Theorem

Let $f \in B^\alpha_\infty(L^p)$ with $\alpha > 0$ and $0 < p \leq \infty$. Then

$$E(f, \Phi_n^{\mathcal{F}})_{L^p} \leq C n^{-\tilde{\alpha}/d} |f|_{B^\alpha_\infty(L^p)}$$

for arbitrary $\tilde{\alpha} < \alpha$.

- Tensor networks achieve (near to) optimal performance for any Besov regularity order (measured in L^p norm).
- They perform as well as optimal linear approximation tools (e.g. splines), without requiring to adapt the tool to the regularity order α.
- The depth (resolution L) of the network is crucial to capture extra regularity.

Now consider the much harder problem of approximating functions from Besov spaces $B^\alpha_q(L^\tau)$ where regularity is measured in a L^τ-norm weaker than L^p-norm.

From results on best n-term approximation using dilated splines, we obtain

Theorem

Let $f \in B^\alpha_q(L^\tau)$ with $\alpha > 0$, $0 < q \le \tau < p < \infty$, $1 \le p < \infty$ and

$$\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}.$$

Then

$$E(f, \Phi^{\mathcal{S}}_n)_{L^p} \le C n^{-\alpha'/d} |f|_{B^\alpha_q(L^\tau)}, \quad E(f, \Phi^{\mathcal{F}}_n)_{L^p} \le C n^{-\alpha'/(2d)} |f|_{B^\alpha_q(L^\tau)},$$

for arbitrary $\alpha' < \alpha$.

- Sparse tensor networks achieve arbitrarily close to optimal rates in $O(n^{-\alpha/d})$ for functions with any Besov smoothness α (measured in L^τ norm), without the need to adapt the tool to the regularity order α.
- Here depth and sparsity are crucial for obtaining near to optimal performance.
- Full tensor networks have slightly lower performance in $O(n^{-\alpha/(2d)})$.

Analytic functions

For function $f : [0,1]$ with analytic extension on an open complex domain

$$D_\rho = \{z \in \mathbb{C} : dist(z, [0,1])) < \frac{\rho - 1}{2}\}, \quad \rho > 1,$$

we obtain an exponential convergence

$$E(f, \Phi_n^{\mathcal{F}})_{L^\infty} \leq C\gamma^{-n^{1/3}},$$

with $\gamma = \min\{\rho, b^{(m+1)/b}\}$.

The proof relies on the approximation of analytic functions with polynomials and the encoding of polynomials with tree tensor networks: a chebychev polynomial p of deree $\bar{m}$ is such that

$$\|f - p\|_{L^\infty} \leq \frac{2}{\rho - 1}\|f\|_{L^\infty(D_\rho)}\rho^{-\bar{m}}$$

A polynomial of degree $\bar{m}$ can be approximated by φ in $\Phi_{L,r,m}$ with an error in $O(b^{-L(m+1)})$, so that

$$\|f - \varphi\|_{L^\infty} \lesssim \rho^{-\bar{m}} + b^{-L(m+1)}$$

We obtain the result by choosing $\bar{m} \sim n^{1/3}$ and $L \sim b^{-1}n^{1/3}$, so that $compl_{\mathcal{F}}(\varphi) \leq n$.

Functions with singularities

Consider the approximation $u(x) = x^\alpha$, $0 < \alpha \leq 1$, in L^∞.

- Piecewise constant linear approximation.

$$u \in B^\alpha_\infty(L^\infty), \quad u \notin B^\beta_\infty(L^\infty) \quad \text{for } \beta > \alpha,$$

  and a piecewise constant approximation on a uniform mesh with n elements gives a convergence in $O(n^{-\alpha})$ in L^∞,

- Piecewise constant nonlinear approximation.

$$u \in BV \subset B^1_\infty(L^1),$$

  and a piecewise constant approximation on an optimal mesh with n elements gives a convergence in $O(n^{-1})$ in L^∞,

- Piecewise constant approximation and tensor networks.
  A piecewise constant approximation on a uniform mesh with 2^d elements exploiting low-rank structures gives an exponential convergence in $O(\beta^{-n})$, where n is the complexity of the representation. Achieves the performance of h-p methods.

High-dimensional approximation

- For Besov spaces $B^{\alpha}_{q}(L^{p})$, tensor networks achieve (near to) optimal rate in $O(n^{-\alpha/d})$ which deteriorates with d, that is the curse of dimensionality.
- For Besov spaces with mixed smoothness $MB^{\alpha}_{q}(L^{p})$, sparse tensor networks achieve near to optimal performance in $O(n^{-\alpha}\log(n)^{d})$. But still the curse of dimensionality.
- For Besov spaces with anisotropic smoothness $AB^{\boldsymbol{\alpha}}_{q}(L^{p})$, sparse tensor networks also achieve near to optimal rates in $O(n^{-s(\boldsymbol{\alpha})/d})$ with

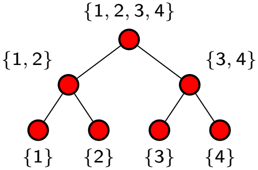
$$s(\boldsymbol{\alpha})/d = (\alpha_1^{-1} + \ldots + \alpha_d^{-1})^{-1}$$

  the aggregated smoothness. Curse of dimensionality can be circumvented with sufficient anisotropy.
- Curse of dimensionality can be circumvented for non usual function classes such as compositions of smooth functions (see Bachmayr, Nouy and Schneider 2021).

Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Assuming that the functions $f_\alpha \in W^{k,\infty}$ with $\|f_\alpha\|_{L^\infty} \leq 1$ and $\|f_\alpha\|_{W^{k,\infty}} \leq B$, the complexity to achieve an accuracy ϵ

$$C(\epsilon) \lesssim \epsilon^{-3/k}(L+1)^3 B^{3L} d^{1+3/2k}$$

with $L = \log_2(d)$ for a balanced tree and $L+1 = d$ for a linear tree.

- Bad influence of the depth through the norm B of functions f_α (roughness).
- For a balanced tree, complexity scales polynomially in d: no curse of dimensionality !
- For $B \leq 1$ (and even for 1-Lipschitz functions), the complexity only scales polynomially in d whatever the tree: no curse of dimensionality !

Outline

Canonical versus tree-based format

Consider a finite dimensional tensor space $V = V^1 \otimes \ldots \otimes V^d$ with $\dim(V_\nu) = \mathbb{R}^N$, which is identified with $\mathbb{R}^{N\times\cdots\times N}$. Denote by $\mathcal{T}_r^T = \{v : \mathrm{rank}_\alpha(v) \leq r, \alpha \in T\}$.

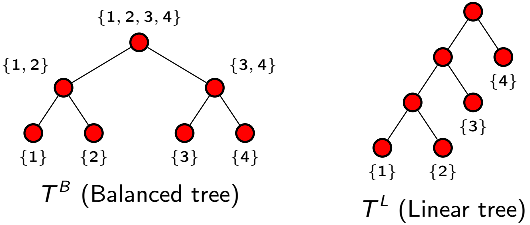
- From canonical format to tree-based format.
  For any v in V and any $\alpha \subset D$, the α-rank is bounded by the canonical rank:
  $$\mathrm{rank}_\alpha(v) \leq \mathrm{rank}(v).$$
  Therefore, for any tree T,
  $$\mathcal{R}_r \subset \mathcal{T}_r^T,$$
  so that an element in $\mathcal{R}_r$ with storage complexity $O(dNr)$ admits a representation in $\mathcal{T}_r^T$ with a storage complexity $O(dNr + dr^{s+1})$ where s is the arity of the tree T.

- From tree-based format to canonical format. For a balanced or linear binary tree, the subset
  $$S = \{v \in \mathcal{T}_r^T : \mathrm{rank}(v) < q^{d/2}\}, \quad q = \min\{N, r\},$$
  is of Lebesgue measure 0.

  Then a typical element $v \in \mathcal{T}_r^T$ with storage complexity of order $dNr + dr^3$ admits a representation in canonical format with a storage complexity of order $dNq^{d/2}$.

Influence of the tree

- For some functions, the choice of tree is not crucial. For example, an additive function
$$u_1(x_1) + \ldots + u_d(x_d)$$
has α-ranks equal to 2 whatever $\alpha \subset D$.
- But usually, different trees lead to different complexities of representations.



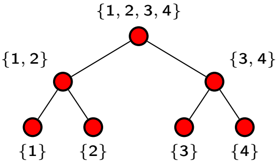
  - If $\text{rank}_{T^L}(u) \leq r$ then $\text{rank}_{T^B}(u) \leq r^2$
  - If $\text{rank}_{T^B}(u) \leq r$ then $\text{rank}_{T^L}(u) \leq r^{\log_2(d)/2}$

Influence of the tree

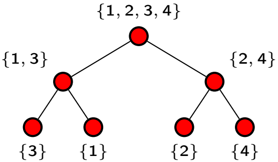
Given a tree T and a permutation σ of $D = \{1, \ldots, d\}$, we define a tree T_σ

$$T_\sigma = \{\sigma(\alpha) : \alpha \in T\}$$

having the same structure as T but different nodes.



T



T_σ with $\sigma = (3, 1, 2, 4)$

If $\text{rank}_T(u) \leq r$ then $\text{rank}_{T_\sigma}(u)$ typically depends on d.

- Consider the Henon-Heiles potential

$$u(x) = \frac{1}{2}\sum_{i=1}^{d} x_i^2 + 0.2\sum_{i=1}^{d-1}(x_i x_{i+1}^2 - x_i^3) + \frac{0.2^2}{16}\sum_{i=1}^{d-1}(x_i^2 + x_{i+1}^2)^2$$

  Using a linear tree $T = \{\{1\},\{2\},\ldots,\{d\},\{1,2\},\{1,2,3\},\ldots,\{1,\ldots,d-1\},D\}$,

$$\mathrm{rank}_T(u) \leq 4, \quad storage(u) = O(d)$$

  but for the permutation

$$\sigma = (1,3,\ldots,d-1,2,4,\ldots,d) \tag{$\star$}$$

  and the corresponding linear tree T_σ,

$$\mathrm{rank}_{T_\sigma}(u) \leq 2d+1, \quad storage(u) = O(d^3).$$

- For a typical tensor in $\mathcal{T}_r^T$ with T a binary tree, its representation in tree based format with tree T_σ, with σ as in ($\star$), has a complexity scaling exponentially with d.

Influence of the tree

- Consider the probability distribution $f(x) = \mathbb{P}(X = x)$ of a Markov chain $X = (X_1, \dots, X_d)$ given by

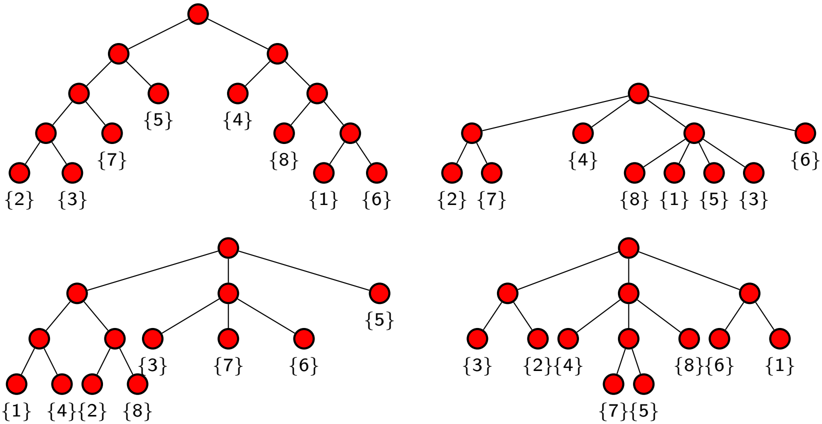
$$f(x) = f_1(x_1) f_{2|1}(x_2|x_1) \dots f_{d|d-1}(x_d|x_{d-1})$$

where bivariate functions $f_{i|i-1}$ have a rank r.

  - With the linear tree T containing interior nodes $\{1,2\}, \{1,2,3\}, \dots, \{1,\dots,d-1\}$, f admits a representation in tree-based format with storage complexity in r^4.
  - The canonical rank of f is exponential in d.
  - But when considering the linear tree T_σ obtained by applying permutation $\sigma = (1,3,\dots,d-1,2,4,\dots,d)$ to the tree T, the storage complexity in tree-based format is also exponential in d.

How to choose a good tree ?

A combinatorial problem...



Outline

Properties of tree tensor networks

We here consider approximation tools based on tensor networks with tensorized functions (with or without sparsity).

They satisfy

(P1) $\Phi_0 = \{0\}$, $0 \in \Phi_n$

(P2) $a\Phi_n = \Phi_n$ for any $a \in \mathbb{R} \setminus \{0\}$ (cone)

(P3) $\Phi_n \subset \Phi_{n+1}$ (nestedness)

(P4) $\Phi_n + \Phi_n \subset \Phi_{cn}$ for some constant c (not too nonlinear)

For $X = L^p$, they further satisfy

(P5) $\bigcup_n \Phi_n$ is dense in L^p for $0 < p < \infty$ (universality),

(P6) for each $f \in L^p$ for $0 < p \leq \infty$, there exists a best approximation in Φ_n (proximinal sets).

Approximation classes

For an approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$, we define for any $\alpha > 0$ the approximation class

$$A^\alpha_\infty(L^p) := A^\alpha_\infty(L^p, \Phi)$$

of functions $f \in L^p$ such that

$$E(f, \Phi_n)_{L^p} \leq Cn^{-\alpha}$$

- Properties (P1)-(P4) of Φ imply that $A^\alpha_\infty(L^p)$ is a quasi-Banach spaces with quasi-semi-norm

$$|f|_{A^\alpha_\infty} := \sup_{n\geq 1} n^\alpha E(f, \Phi_n)_{L^p}$$

- Full and sparse complexity measures yield two different approximation spaces

$$\mathcal{F}^\alpha_\infty(L^p) = A^\alpha_\infty(L^p, \Phi^{\mathcal{F}}), \quad \mathcal{S}^\alpha_\infty(L^p) = A^\alpha_\infty(L^p, \Phi^{\mathcal{S}})$$

  such that

$$\mathcal{F}^\alpha_\infty(L^p) \hookrightarrow \mathcal{S}^\alpha_\infty(L^p) \hookrightarrow \mathcal{F}^{\alpha/2}_\infty(L^p)$$

From results on the approximation properties for Besov spaces, we have the following results.

- Let $\alpha > 0$ and $0 < p \leq \infty$. For arbitrary $\tilde{\alpha} < \alpha$,
$$B^{\alpha}_{q}(L^p) \hookrightarrow \mathcal{F}^{\tilde{\alpha}/d}_{q}(L^p)$$
and
$$MB^{\alpha}_{q}(L^p) \hookrightarrow \mathcal{S}^{\tilde{\alpha}}_{q}(L^p).$$
For arbitrary $\tilde{s} < s(\boldsymbol{\alpha}) := d(\alpha_1^{-1} + \ldots + \alpha_d^{-1})^{-1}$,
$$AB^{\boldsymbol{\alpha}}_{q}(L^p) \hookrightarrow \mathcal{S}^{\tilde{s}/d}_{q}(L^p)$$
- For $\alpha > 0$, $1 \leq p < \infty$, $0 < q \leq \tau < p < \infty$ and $\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}$,
$$B^{\alpha}_{q}(L^{\tau}) \hookrightarrow \mathcal{S}^{\tilde{\alpha}/d}_{\infty}(L^p) \hookrightarrow \mathcal{F}^{\tilde{\alpha}/(2d)}_{\infty}(L^p)$$
for arbitrary $\tilde{\alpha} < \alpha$, and similar results for anisotropic and mixed smoothness.

No inverse embedding

For any $\alpha > 0$, $q \leq \infty$, and any β,

$$\mathcal{F}^{\alpha}_{\infty}(L^p) \not\hookrightarrow B^{\beta}_{\infty}(L^p).$$

That means that approximation classes contain functions that have no smoothness in a classical sense.

Tensor networks may be useful for the approximation of functions beyond standard smoothness classes.

Open questions

- What are the properties of the approximation tool with free tree

$$\Phi_n = \{f \in \Phi_{L,T_L,r} : L \in \mathbb{N}_0, T_L \subset 2^{\{1,\dots,(L+1)d\}}, r \in \mathbb{N}^{\#T}, compl(f) \leq n\}$$

  Higher expressivity (or larger approximation classes) but how much higher ?

- What about expressivity and approximation classes of more general tensor networks ?

References I

M. Ali and A. Nouy.
Approximation with tensor networks. part i: Approximation spaces.
ArXiv, abs/2007.00118, 2020.

M. Ali and A. Nouy.
Approximation with tensor networks. part ii: Approximation rates for smoothness classes.
ArXiv, abs/2007.00128, 2020.

M. Ali and A. Nouy.
Approximation with tensor networks. part iii: Multivariate approximation.
ArXiv, abs/2007.00128, 2020.

M. Bachmayr, A. Nouy and R. Schneider.
Approximation power of tree tensor networks for compositional functions.
In preparation.

R. A. DeVore and G. G. Lorentz.
Constructive approximation, volume 303.
Springer Science & Business Media, 1993.

L. Grasedyck.
Polynomial approximation in hierarchical Tucker format by vector-tensorization.
Inst. für Geometrie und Praktische Mathematik, 2010.

V. Kazeev and C. Schwab.
Approximation of singularities by quantized-tensor fem.
PAMM, 15(1):743–746, 2015.

V. Kazeev, I. Oseledets, M. Rakhuba, and C. Schwab.
Qtt-finite-element approximation for multiscale problems i: model problems in one dimension.
Advances in Computational Mathematics, 43(2):411–442, Apr 2017.

R. Schneider and A. Uschmajew.
Approximation rates for the hierarchical tensor format in periodic sobolev spaces.
Journal of Complexity, 30(2):56 – 71, 2014.

CEMRACS,
July 19-23, 2021

Approximation and learning with tensor networks

Part III: Computational aspects

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Computing with tensor networks

We here present some algorithms for the approximation of tensors (or functions) using tensor networks.

Different contexts depending on the available information on the tensor:

- all entries of the tensor,
- equations satisfied by the tensor,
- some entries, either arbitrary or structured,
- more general functionals of the tensor.

Available software

- tensap. A Python package for the approximation of functions and tensors. (link to GitHub page).
- ApproximationToolbox. An object-oriented MATLAB toolbox for the approximation of functions and tensors. (link to GitHub page).

Outline

Outline

Hilbertian setting

We consider a tensor u in a Hilbert tensor space $V^1 \otimes \ldots \otimes V^d$ and we assume that u is given as a full tensor or in a certain low-rank format.

We present truncation schemes for finding a low-rank approximation of u with reduced complexity, relying on the standard singular value decomposition of order-two tensors.

We denote by $\|\cdot\|$ the canonical norm on $V^1 \otimes \ldots \otimes V^d$.

For an algebraic tensor in $\mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_d}$, $\|\cdot\|$ is the Frobenius norm

$$\|u\|^2 = \sum_{i_1 \in I_1} \ldots \sum_{i_d \in I_d} u(i_1, \ldots, i_d)^2$$

Truncated singular value decomposition for order-two tensors

An order-two tensor u in $V^1 \otimes V^2$ admits a singular value decomposition

$$u = \sum_{k \geq 1} \sigma_k v_k^1 \otimes v_k^2,$$

where the singular values $\sigma(u) = \{\sigma_k\}_{k \geq 1}$ are sorted by decreasing order.

An element of best approximation of u in the set of tensors with rank bounded by r is provided by the truncated singular value decomposition

$$u_r = \sum_{k=1}^{r} \sigma_k v_k^1 \otimes v_k^2,$$

with an error

$$\|u - u_r\|^2 = \min_{\operatorname{rank}(v) \leq r} \|u - v\|^2 = \sum_{k \geq r+1} \sigma_k^2.$$

Truncated singular value decomposition for order-two tensors

An approximation u_r with relative precision ϵ, such that

$$\|u - u_r\| \leq \epsilon \|u\|,$$

can be obtained by choosing a rank r such that

$$\sum_{k \geq r+1} \sigma_k^2 \leq \epsilon^2 \sum_{k \geq 1} \sigma_k^2.$$

The complexity of computing the singular value decomposition of a tensor u is $O(n^3)$ if $\dim(V^1) = \dim(V^2) = n$. If u is given in low-rank format $u = \sum_{k=1}^{R} a_k \otimes b_k$, with a rank $R < n$, the complexity breaks down to $O(R^3 + 2Rn^2)$.

Higher-order singular value decomposition

For a non-empty subset α in $D = \{1, \ldots, d\}$, a tensor $u \in V^1 \otimes \ldots \otimes V^d$ can be identified with its matricisation

$$\mathcal{M}_\alpha(u) \in V^\alpha \otimes V^{\alpha^c},$$

an order-two tensor which admits a singular value decomposition

$$\mathcal{M}_\alpha(u) = \sum_{k \geq 1} \sigma_k^\alpha v_k^\alpha \otimes w_k^{\alpha^c} \equiv u.$$

$\sigma^\alpha(u) := \{\sigma_k^\alpha\}_{k \geq 1}$ are the α-singular values of u.

The α-rank of u is the number of non-zero α-singular values

$$\mathrm{rank}_\alpha(u) = \|\sigma^\alpha(u)\|_0.$$

Higher-order singular value decomposition

By sorting the α-singular values by decreasing order, an approximation u_r with α-rank r can be obtained by retaining the r largest α-singular values, i.e.

$$u_r \equiv \sum_{k=1}^{r} \sigma_k^\alpha v_k^\alpha \otimes w_k^{\alpha^c},$$

The vectors $\{v_1^\alpha, \dots, v_{r_\alpha}^\alpha\}$ are the dominant α-singular vectors of u or α-principal components of u.

The space $U_{r_\alpha}^\alpha = span\{v_1^\alpha, \dots, v_{r_\alpha}^\alpha\}$ is the dominant α-principal subpace of u.

Denote by $P_{U_{r_\alpha}^\alpha}$ the orthogonal projection from V^α to $U_{r_\alpha}^\alpha$ and by $\mathcal{P}_{U_{r_\alpha}^\alpha} = P_{U_{r_\alpha}^\alpha} \otimes id_{\alpha^c}$ the orthogonal projection defined on V such that for $v^\alpha \otimes w^{\alpha^c} \in V^\alpha \otimes V^{\alpha^c}$,

$$\mathcal{P}_{U_{r_\alpha}^\alpha}(v^\alpha \otimes w^{\alpha^c}) = (P_{U_{r_\alpha}^\alpha} v^\alpha) \otimes w^{\alpha^c}$$

We have

$$u_r = \mathcal{P}_{U_{r_\alpha}^\alpha} u$$

and

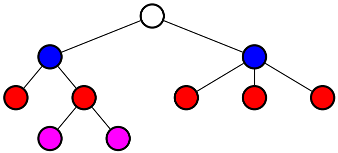
$$\|u - u_r\|^2 = \min_{\text{rank}_\alpha(v) \leq r} \|u - v\|^2 = \sum_{k>r} (\sigma_k^\alpha)^2.$$

Truncation scheme for tree-based tensor formats

For tree-based tensor formats

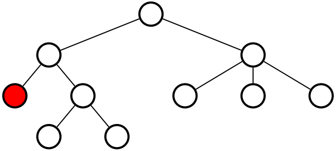
$$\mathcal{T}_r^T(V) = \{v \in V : \mathrm{rank}_\alpha(v) \leq r_\alpha, \alpha \in T\},$$

where T is a dimension partition tree over $D = \{1, \ldots, d\}$, different variants of higher order singular value decomposition (also called hierarchical singular value decomposition) can be defined from singular value decompositions of matricisations $\mathcal{M}_\alpha(u)$ of a tensor u.



Leaves to root truncation scheme for tree-based tensor formats

For each leaf node α, let $U^{\alpha}_{r_\alpha}$ be the r_α-dimensional α-principal subspace of u.

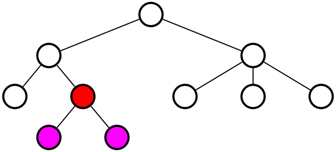


For each interior node $\alpha \in T \setminus \{D\}$ with children $S(\alpha)$, define a tensor space

$$V_\alpha = \bigotimes_{\beta \in S(\alpha)} U^{\beta}_{r_\beta}$$

and let $U^{\alpha}_{r_\alpha} \subset V_\alpha$ be the r_α-dimensional α-principal subspace of

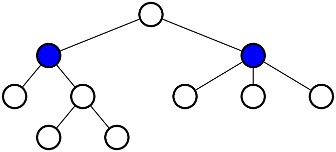
$$u_\alpha = \mathcal{P}_{V_\alpha} u$$



Leaves to root truncation scheme for tree-based tensor formats

Finally define u_r as the orthogonal projection onto the tensor space $V_D = \bigotimes_{\alpha \in S(D)} U_\alpha$

$$u_r = \mathcal{P}_r^{(1)} u = \mathcal{P}_r^{(1)} \dots \mathcal{P}_r^{(L)} u$$



The obtained approximation u_r is such that

$$\|u - u_r\|^2 \leq \sum_{\alpha \in T \setminus D} \min_{\mathrm{rank}_\alpha(v) \leq r_\alpha} \|u - v\|^2 = \sum_{\alpha \in T \setminus D} \sum_{k_\alpha > r_\alpha} (\sigma^\alpha_{k_\alpha})^2,$$

from which we deduce that u_r is a quasi-optimal approximation of u in $\mathcal{T}_r^T$ such that

$$\|u - u_r\| \leq C(T) \min_{v \in \mathcal{T}_r^T} \|u - v\|,$$

where $C(T) = \sqrt{\#T - 1}$ is the square root of the number of projections applied to the tensor. The number of nodes of a dimension partition tree T being bounded by $2d - 1$,

$$C(T) \leq \sqrt{2d - 2}.$$

Also, if we select the ranks $(r_\alpha)_{\alpha \in T \setminus D}$ such that for all α

$$\sum_{k_\alpha > r_\alpha} (\sigma^\alpha_{k_\alpha})^2 \leq \frac{\epsilon^2}{C(T)^2} \sum_{k_\alpha \geq 1} (\sigma^\alpha_{k_\alpha})^2 = \frac{\epsilon^2}{C(T)^2} \|u\|^2,$$

we finally obtain an approximation u_r with relative precision ϵ,

$$\|u - u_r\| \leq \epsilon \|u\|.$$

Leaves to root truncation scheme for tree-based tensor formats

If u is in some tensor space $W = W_1 \otimes \ldots \otimes W_d$ and $V = V_1 \otimes \ldots \otimes V_d$ is a finite-dimensional tensor subspace of W, an approximation in the tensor format $\mathcal{T}_r^T(V)$ can be obtained by modifying the procedure for the leaves.

For each leaf node α, $U^\alpha_{r_\alpha}$ is defined as a α-principal subspace of $u_\alpha = \mathcal{P}_{V_\alpha} u$.

Theorem (Fixed rank)

For a given T-rank, we obtain an approximation $u_r \in \mathcal{T}_r^T(V)$ such that

$$\|u_r - u\|^2 \leq C(T)^2 \min_{v \in \mathcal{T}_r^T} \|v - u\|^2 + \sum_{\text{leaves } \alpha} \|u - \mathcal{P}_{V_\alpha} u\|^2$$

Theorem (Fixed precision)

For a desired precision ϵ, if the α-ranks are determined such that

$$\|\mathcal{P}_{U^\alpha_{r_\alpha}} u_\alpha - u_\alpha\| \leq \frac{\epsilon}{C(T)} \|u_\alpha\|,$$

we obtain an approximation u_r such that

$$\|u_r - u\|^2 \leq \epsilon^2 \|u\|^2 + \sum_{\text{leaves } \alpha} \|u - \mathcal{P}_{V_\alpha} u\|^2.$$

Efficient truncation algorithms

Recent works for efficient truncation algorithms

- Randomized linear algebra [Che/Wei'19,Sun'20,Huber'17]
- Block-wise tensor compressions [Ehrlacher'21]
- Parallel algorithms [Grigori/Kumar'20,Daas'20]
- ...

Outline

Learning from structured evaluations

For the approximation of a tensor (or function) in tree-based format from evaluations of the tensor at some entries, different strategies have been proposed, either based on cross approximation [Oseledets'10, Ballani'13] or principal component analysis [Nouy'19, Haberstich'21].

These methods rely on structured evaluations

$$u(x_\alpha^i, x_{\alpha^c}^j)$$

where x_α^i are samples of the variables x_α, and $x_{\alpha^c}^j$ samples of the variables x_{α^c}.

Learning from principal component analysis

Assume that $X = (X_1, \ldots, X_d)$ has a probability measure $\mu = \mu_1 \otimes \ldots \otimes \mu_d$ with support $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$.

Consider a multivariate function $u \in L^2_\mu(\mathcal{X})$ and assume that we can evaluate the function for arbitrary instance x of X.

For each a subset of variables α and its complementary subset $\alpha^c = D \setminus \alpha$, u is identified with a bivariate function which admits a singular value decomposition

$$u(x_\alpha, x_{\alpha^c}) = \sum_{k=1}^{\mathrm{rank}_\alpha(u)} \sigma_k^\alpha v_k^\alpha(x_\alpha) v_k^{\alpha^c}(x_{\alpha^c})$$

Learning from principal component analysis

The subspace of α-principal components

$$U_\alpha = span\{v_1^\alpha, \dots, v_{r_\alpha}^\alpha\}$$

is such that

$$u_{r_\alpha}(\cdot, x_{\alpha^c}) = \mathcal{P}_{U_\alpha} u(\cdot, x_{\alpha^c})$$

It is solution of

$$\min_{\dim(U_\alpha)=r_\alpha} \|u - \mathcal{P}_{U_\alpha} u\|^2$$

that is for $\|\cdot\|$ the $L^2_\mu(\mathcal{X})$-norm,

$$\min_{\dim(U_\alpha)=r_\alpha} \mathbb{E}\left(\|u(\cdot, X_{\alpha^c}) - \mathcal{P}_{U_\alpha} u(\cdot, X_{\alpha^c})\|^2_{L^2_{\mu_\alpha}(\mathcal{X}_\alpha)}\right)$$

where u is seen as a function-valued random variable

$$u(\cdot, X_{\alpha^c}) \in L^2_{\mu_\alpha}(\mathcal{X}_\alpha).$$

Learning algorithm based on principal component analysis

In order to construct an approximation in the tree-based format $\mathcal{T}_r^T(V)$, with V some feature tensor space, we apply the root to leaves procedure.

For a feasible algorithm using samples:

- Replacement of orthogonal projections by sampled-based projections.
- Statistical estimation of principal subspaces.

From orthogonal to sampled-based projections

Orthogonal projections $\mathcal{P}_{V_\alpha}$ on subspaces V_α are replaced by oblique projections $\mathcal{I}_{V_\alpha}$ using samples, typically interpolation or least-squares projection.

For a function u and a given value x_{α^c} of the group of variables X_{α^c},

$$\mathcal{I}_{V_\alpha} u(\cdot, x_{\alpha^c}) = \sum_{i=1}^{M_\alpha} a_i(x_{\alpha^c}) \psi_i^\alpha(\cdot)$$

where the ψ_i^α form a basis of V_α, and the coefficients $a_i(x_{\alpha^c})$ depend on evaluations $u(x_\alpha^k, x_{\alpha^c})$ for some samples x_α^k of X_α (interpolation points or random samples).

In practice,

- for interpolation, possible use of magic points x_α^i [Nouy '19],
- for least-squares projection, possible use of optimal weighted least-squares for a control of the norm of operators $\mathcal{I}_{V_\alpha}$ [Cohen/Migliorati'17,Habertisch '21].

Statistical estimation of principal subspaces

The α-principal subspaces U_α of $u_\alpha = \mathcal{I}_{V_\alpha} u$ are defined by

$$\min_{\dim(U_\alpha)=r_\alpha} \mathbb{E}\left(\|\mathcal{I}_{V_\alpha} u(\cdot, X_{\alpha^c}) - \mathcal{P}_{U_\alpha}\mathcal{I}_{V_\alpha} u(\cdot, X_{\alpha^c})\|^2_{L^2_{\mu_\alpha}(\mathcal{X}_\alpha)}\right)$$

Principal subspaces can be estimated using i.i.d. samples $u(\cdot, x^j_{\alpha^c})$ of this random variable and by solving

$$\min_{\dim(U_\alpha)=r_\alpha} \frac{1}{N_\alpha}\sum_{j=1}^{N_\alpha} \|\mathcal{I}_{V_\alpha} u(\cdot, x^j_{\alpha^c}) - \mathcal{P}_{U_\alpha}\mathcal{I}_{V_\alpha} u(\cdot, x^j_{\alpha^c})\|^2_{L^2_{\mu_\alpha}(\mathcal{X}_\alpha)}$$

where $\{x^j_{\alpha^c}\}_{j=1}^{N_\alpha}$ are i.i.d. samples of the group of variables X_{α^c}.

If the projection $\mathcal{I}_{V_\alpha}$ is based on a set of M_α samples of X_α, this requires the evaluation of u at the $M_\alpha \times N_\alpha$ points

$$\{(x^i_\alpha, x^j_{\alpha^c}) : 1 \le i \le M_\alpha, 1 \le j \le N_\alpha\}.$$

Outline

Direct optimization in subsets of tensor networks

Consider a subset of tensors $\mathcal{M}_r$ that admits a multilinear parametrization of the form

$$v(x_1, \dots, x_d) = \sum_{k_1=1}^{r_1} \dots \sum_{k_L=1}^{r_L} \prod_{\nu=1}^{d} v^{(\nu)}(x_\nu, (k_i)_{i \in S_\nu}) \prod_{\nu=d+1}^{M} v^{(\nu)}((k_i)_{i \in S_\nu})$$

where $\boldsymbol{v} = \{v^{(\nu)}\}_{\nu=1}^{M}$ is a tensor network, and each tensor $v^{(\nu)}$ is in a space $P^{(\nu)}$.

We have

$$\mathcal{M}_r = \{v = \Psi(v^{(1)}, \dots, v^{(M)}) : v^{(\nu)} \in P^{(\nu)}, 1 \leq \nu \leq M\},$$

where Ψ is a multilinear map.

The problem

$$\min_{v \in \mathcal{M}_r} \mathcal{J}(v)$$

can be written as an optimization problem over the parameters

$$\min_{v^{(1)}} \dots \min_{v^{(M)}} \mathcal{J}(\Psi(v^{(1)}, \dots, v^{(M)})).$$

Alternating minimization algorithm

The alternating minimization algorithm consists in solving successively minimization problems

$$\min_{v^{(\nu)} \in P^{(\nu)}} \mathcal{J}(\Psi(v^{(1)}, \dots, v^{(\nu)}, \dots, v^{(M)})) := \min_{v^{(\nu)} \in P^{(\nu)}} \mathcal{J}_\nu(v^{(\nu)}) \tag{1}$$

over the parameter $v^{(\nu)}$, letting the other parameters $v^{(\eta)}$, $\eta \neq \nu$, fixed.

When $P^{(\nu)}$ is a linear vector space, problem (1) is a linear approximation problem.

If $\mathcal{J}$ is a convex (resp. differentiable) functional, then $\mathcal{J}_\nu$ is a convex (resp. differentiable) functional.

Direct optimization in subsets of tensor networks

Other optimization algorithms (e.g. gradient descent, Newton) can be used, possibly exploiting the geometry of tree tensor networks manifolds.

Under rather standard assumptions, some results have been obtained for the convergence of algorithms: local convergence to a global optimizer, or global convergence to stationary points.

But no guaranty for obtaining a global optimizer of a general (even convex) functional in subsets of tensor networks (NP-hard problem).

Rank adaptation

For the adaptation of ranks, different strategies have been proposed:

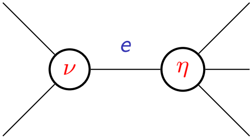
- Modified alternating minimization algorithms [Holtz et al '12] or DMRG, where rank adaptation is performed during optimization,
- Alternating minimal energy methods [Dolgov et al '14], where optimization is also combined with rank adaptation,
- Optimization in a subset with fixed rank followed by rank adaptation [Grelier/Nouy/Chevreuil'18, Grelier/Nouy/Lebrun'19,Grasedyck/Kramer '19]

Modified alternating minimization algorithm

Modified alternating minimization algorithm[1] is a modification of the alternating minimization algorithm which allows for an rank adaptation "on the fly".

It can be used for optimization with tree tensor nteworks or more general tensor networks.

At each step of the algorithm, we consider two nodes ν and η connected by an edge e and we update simultaneously the associated parameters $p^{(\nu)}$ and $p^{(\eta)}$.



[1] known as DMRG algorithm (for Density Matrix Renormalization Group) for tensor networks.

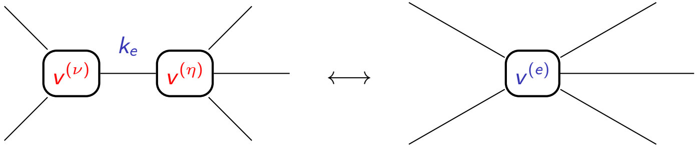
Modified alternating minimization algorithm

In the expression of a tensor $v = \Psi(v^{(1)}, \dots, v^{(M)})$, the two tensors $v^{(\nu)}$ and $v^{(\eta)}$ connected by the edge e appear as

$$\sum_{k_e=1}^{r_e} v^{(\nu)}(k_e, \ldots) v^{(\eta)}(k_e, \ldots) := v^{(e)}(\ldots)$$

where $v^{(e)}$ is a tensor of order

$$\text{order}(v^{(e)}) = \text{order}(v^{(\nu)}) + \text{order}(v^{(\eta)}) - 2.$$



This corresponds to a new tensor networks where the nodes ν and η and edge e are replaced by a single node e, and a new parametrization

$$v = \Psi^e(\ldots, v^{(e)}, \ldots).$$

Modified alternating minimization algorithm

We first solve an optimization problem

$$\min_{v^{(e)}} \mathcal{J}(\Psi^e(\ldots, v^{(e)}, \ldots))$$

for obtaining an new value of the tensor $v^{(e)}$.

Then, we compute a low-rank approximation of the tensor $v^{(e)}$

$$v^{(e)}(...) \approx \sum_{k_e=1}^{r_e} v^{(\nu)}(k_e, ...)v^{(\eta)}(k_e, ...)$$

where the rank r_e in general differs from the initial rank.

In practice, the approximation is obtained using truncated singular value decomposition.

Outline

Iterative methods with tensor truncation

Another strategy for solving an operator equation

$$Au = b$$

or a more general optimization problem

$$\min_{v \in V} \mathcal{J}(v)$$

is to rely on classical iterative methods by interpreting all standard algebraic operations on vector spaces as algebraic operations in tensor spaces.

As a motivating example, consider a simple Richardson algorithm

$$u^n = u^{n-1} - \omega(Au^{n-1} - b).$$

For A and b given in tensor formats, computing u^n involves standard algebraic operations.

However, the representation rank of the iterates dramatically increases since

$$\mathrm{rank}(u^n) \approx \mathrm{rank}(A)\,\mathrm{rank}(u^{n-1}) + \mathrm{rank}(u^{n-1}) + \mathrm{rank}(b).$$

This requires additional truncation steps for reducing the ranks of the iterates, such as

$$u^n = T(u^{n-1} - \omega(Au^{n-1} - b)),$$

where $T(v)$ provides a low-rank approximation of v.

We now analyze the behavior of these algorithms depending on the properties of the truncation operator T.

Fixed point iterations algorithm

Let us consider a problem which can be written as a fixed point problem

$$F(u) = u,$$

where $F : V \to V$ is a contractive map, such that for all $u, v \in V$,

$$\|F(u) - F(v)\| \leq \rho \|u - v\|,$$

with $0 \leq \rho < 1$.

Then, consider the fixed point iterations algorithm

$$u^{n+1} = F(u^n)$$

which provides a sequence $(u^n)_{n \geq 1}$ which converges to u, such that

$$\|u - u^n\| \leq \rho^n \|u - u^0\|.$$

Example

For a problem $Au = b$, consider $F(u) = u - \omega(Au - b)$, with ω such that $\|I - \omega A\| < 1$.
Fixed point iterations $u^{n+1} = u^n - \omega(Au^n - b)$ correspond to Richardson iterations.

Perturbed fixed point iterations algorithm

Now consider the perturbed fixed point iterations

$$v^{n+1} = F(u^n), \quad u^{n+1} = T(v^{n+1})$$

where T is a mapping which for a tensor v provides an approximation (called truncation) $T(v)$ in a certain low-rank format $\mathcal{M}_r$.

Truncations with controlled relative precision

Suppose that the mapping T provides an approximation with relative precision ϵ, i.e.

$$\|T(v) - v\| \leq \epsilon \|v\|.$$

This is made possible by using an adaptation of the ranks.

Then the sequence $(u^n)_{n\geq 1}$ is such that

$$\|u - u^n\| \leq \gamma^n \|u - u^0\| + \frac{\epsilon}{1-\gamma}\|u\|,$$

with $\gamma = \rho(1+\epsilon)$. Therefore, if $\gamma < 1$

$$\limsup_{n\to\infty} \|u - u^n\| \leq \frac{\epsilon}{1-\gamma}\|u\|$$

which means that the sequence tends to enter a neighborhood of u with radius $\frac{\epsilon}{1-\gamma}\|u\|$.

The drawback of this algorithm is that the ranks of the iterates are not controlled and may become very high during the iterations.

Truncations in fixed subsets

Now consider that the mapping T provides an approximation in a fixed subset of tensors $\mathcal{M}_r$ with rank bounded by r.

Let us assume that for all v, $T(v)$ provides a quasi-optimal approximation of v such that

$$\|T(v) - v\| \leq C \min_{w \in \mathcal{M}_r} \|v - w\|. \tag{2}$$

A practical realization of a mapping T verifying (2) is provided by truncated higher-order singular value decompositions, where

$$C = O(\sqrt{d}).$$

Truncations in fixed subsets

Let u_r be an element of best approximation of u, with

$$\|u - u_r\| = \min_{v \in \mathcal{M}_r} \|u - v\|.$$

The sequence $(u^n)_{n \geq 1}$ is such that

$$\|u - u^n\| \leq \gamma^n \|u - u^0\| + \frac{C}{1 - \gamma} \|u - u_r\|,$$

with $\gamma = \rho(1 + C)$. If $\gamma < 1$ (which may be quite restrictive on ρ), we obtain

$$\limsup_{n \to \infty} \|u - u^n\| \leq \frac{C}{1 - \gamma} \min_{v \in \mathcal{M}_r} \|u - v\|,$$

which means that the sequence tends to enter a neighborhood of u with radius $\frac{C}{1-\gamma}\sigma_r$, where σ_r is the best approximation error of u by elements of $\mathcal{M}_r$.

An advantage of this approach is that the ranks of the iterates are controlled. A drawback is that the condition $\gamma < 1$ imposes to rely on an iterative method with small contractivity constant $\rho < (1 + C)^{-1}$, which may be quite restrictive (requires good preconditioners).

Truncations with non-expansive maps

Now we assume that the mapping T providing an approximation in low-rank format is non-expansive, i.e.

$$\|T(v) - T(w)\| \leq \|v - w\| \tag{3}$$

The sequence u^n is defined by

$$u^{n+1} = G(u^n),$$

where $G = T \circ F$ is a contractive mapping with the same contractivity constant ρ as F. Therefore, the sequence u^n converges to the unique fixed point $u^\star$ of G such that

$$G(u^\star) = u^\star,$$

with

$$\|u^\star - u^n\| \leq \rho^n \|u^\star - u^0\|.$$

The obtained approximation $u^\star$ is such that

$$(1+\rho)^{-1}\|u - T(u)\| \leq \|u - u^\star\| \leq (1-\rho)^{-1}\|u - T(u)\|.$$

A practical realization of a mapping T verifying (2) is provided by a truncation operator based on soft thresholding of singular values. The ranks of the iterates are not controlled. However, it is observed in practice that the ranks of iterates are usually lower than with truncations with controlled relative precision.

Outline

Thresholding of singular values

Consider an order two tensor u in a Hilbert tensor space $V \otimes W$. equipped with the canonical norm.

Hard thresholding of singular values

The hard singular value thresholding operator $\mathcal{HT}_\tau$ is defined for an order-two tensor u with singular value decomposition $\sum_{k\geq 1} \sigma_k v_k \otimes w_k$ by

$$\mathcal{HT}_\tau(u) = \sum_{k\geq 1} HT_\tau(\sigma_k) v_k \otimes w_k,$$

where $HT_\tau(t) = t\,1_{|t|>\tau}$ is the hard thresholding function such that

$$HT_\tau(\sigma_k) = \begin{cases} \sigma_k & \text{if } \sigma_k > \tau \\ 0 & \text{if } \sigma_k \leq \tau \end{cases}.$$

The error after hard thresholding is

$$\|u - \mathcal{HT}_\tau(u)\|^2 = \sum_{k\geq 1} \sigma_k^2\, 1_{\sigma_k \leq \tau}.$$

$\mathcal{HT}_\tau(u)$ is a solution of the problem

$$\min_v \|u - v\|^2 + \tau^2 \operatorname{rank}(v)$$

where $\operatorname{rank}(v) = \|\sigma(v)\|_0$.

Soft thresholding of singular values

The soft singular value thresholding operator $\mathcal{ST}_\tau$ is defined for a tensor u with singular value decomposition $\sum_{k\geq 1} \sigma_k v_k \otimes w_k$ by

$$\mathcal{ST}_\tau(u) = \sum_{k\geq 1} ST_\tau(\sigma_k) v_k \otimes w_k,$$

where $ST_\tau(t) = (|t| - \tau)_+ \operatorname{sign}(t)$ is the soft thresholding function, such that

$$ST_\tau(\sigma_k) = (\sigma_k - \tau)_+ = \begin{cases} \sigma_k - \tau & \text{if } \sigma_k \geq \tau \\ 0 & \text{if } \sigma_k < \tau \end{cases}.$$

The error after soft thresholding is

$$\|u - \mathcal{ST}_\tau(u)\|^2 = \sum_{k\geq 1} (\sigma_k - (\sigma_k - \tau)_+)^2 = \sum_{\sigma_k \leq \tau} {\sigma_k}^2 + \sum_{\sigma_k > \tau} \tau^2.$$

Soft thresholding of singular values

$\mathcal{ST}_\tau(u)$ is a solution of the problem

$$\min_v \frac{1}{2}\|u-v\|^2 + \tau\|\sigma(v)\|_1$$

where $\|\sigma(v)\|_1$ is the nuclear norm of v, which is a convex regularization of the functional $v \mapsto \mathrm{rank}(v)$.

In convex analysis, $\mathcal{ST}_\tau$ is known as the proximal operator of the convex function $v \mapsto \tau\|\sigma(v)\|_1$.

The operator $\mathcal{ST}_\tau$ is non-expansive, that means for all u, v,

$$\|\mathcal{ST}_\tau(u) - \mathcal{ST}_\tau(v)\| \leq \|u-v\|,$$

which is an important property for the analysis of algorithms with tensor truncations.

Convex relaxation

A general optimization problem over a subset of tensors with bounded rank

$$\min_{\text{rank}(v)\leq r} \mathcal{J}(v)$$

is equivalent to

$$\min_{v} \mathcal{J}(v) + \tau \,\text{rank}(v)$$

for some value of τ.

A convex optimization problem is obtained by replacing $\text{rank}(v) = \|\sigma(v)\|_0$ by the function $\|\sigma(v)\|_1 = \|v\|_*$ (the nuclear norm of v)

$$\min_{v} \mathcal{J}(v) + \tau \|v\|_*$$

Proximal algorithms

Consider the problem

$$\min_v \mathcal{J}(v) + \tau\|v\|_*$$

A proximal algorithm constructs a sequence $(u^n)_{n\geq 1}$ as follows.

At iteration n, we linearize the function $\mathcal{J}$ around u^n and define u^{n+1} as the solution of

$$\min_v \mathcal{J}(u^n) + (\nabla\mathcal{J}(u^n), v - u^n) + \frac{\beta}{2}\|u - u^n\|^2 + \tau\|v\|_*$$

where β is a parameter.

This is equivalent to solving

$$\min_v \frac{1}{2}\|v - (u^n - \beta^{-1}\nabla\mathcal{J}(u^n))\|^2 + \frac{\tau}{\beta}\|v\|_*$$

whose solution is provided by

$$u^{n+1} = \mathrm{ST}_{\tau/\beta}(u^n - \beta^{-1}\nabla\mathcal{J}(u^n))$$

where $\mathrm{ST}_{\tau/\beta}$ is the proximal operator of $v \mapsto \frac{\tau}{\beta}\|v\|_*$.

Hard and soft singular values thresholding for higher order tensors

For a higher order tensor u in a Hilbert tensor space $V = V_1 \otimes \ldots \otimes V_d$, we can naturally define hard and soft singular values thresholding operators $\mathcal{HS}_\tau^\alpha$ and $\mathcal{ST}_\tau^\alpha$ associated with the singular value decomposition of the matricisation $\mathcal{M}_\alpha(u)$ of u.

These operators are such that

$$\mathcal{HS}_\tau^\alpha(u) = \arg\min_v \|u - v\|^2 + \tau^2 \operatorname{rank}_\alpha(v),$$

and

$$\mathcal{ST}_\tau^\alpha(u) = \arg\min_v \frac{1}{2}\|u - v\|^2 + \tau\|\sigma^\alpha(u)\|_1.$$

Hard and soft singular values thresholding for higher order tensors

Hard and soft thresholding operators can then be defined for the approximation in a tree-based format $\mathcal{T}_r^T(V)$, with T a dimension tree (or a subset T of a dimension tree),

Hard and soft thresholding operators $\mathcal{HT}_\tau^T$ and $\mathcal{ST}_\tau^T$ can be respectively defined as compositions of hard and soft thresholding operators (sequence of truncations from the root to the leaves),

$$\mathcal{HT}_\tau^T = \mathcal{HT}_\tau^{\alpha_M} \circ \ldots \circ \mathcal{HT}_\tau^{\alpha_1}$$

and

$$\mathcal{ST}_\tau^T = \mathcal{ST}_\tau^{\alpha_M} \circ \ldots \circ \mathcal{ST}_\tau^{\alpha_1}$$

where the set of nodes $\{\alpha_1, \ldots, \alpha_M\} = T \setminus \{D\}$ is sorted by increasing level.

The soft-thresholding operator $\mathcal{ST}_\tau^T$ is non-expansive, i.e.

$$\|\mathcal{ST}_\tau^T(u) - \mathcal{ST}_\tau^T(v)\| \leq \|u - v\|$$

for all tensors u, v.

See [Rauhut'17] and [Bachmayr'16] for further details and applications to tensor completion and solution of operator equations.

Convex relaxation for tree-based formats

Given a tree-based format $\mathcal{T}_r^T(V)$, a convex relaxation of the problem

$$\min_{v \in \mathcal{T}_r^T(V)} \mathcal{J}(v)$$

can be defined as

$$\min_{v \in V} \mathcal{J}(v) + \tau \sum_{\alpha \in T \setminus \{D\}} \|\sigma^\alpha(u)\|_1. \qquad (\star)$$

- Algorithms based on soft thresholding of singular values appear as specific algorithms for solving the relaxed optimization problem $(\star)$.
- But this relaxation is known to be far from optimal convex relaxation.
- For Tucker tensors, a better convex relaxation is based on tensor nuclear norm [Yuan/Zhang'16].
- Finding a good convex relaxation for general tree-based formats remains an open problem.

References I

- **Higher-order singular value decompositions and related tensor truncation schemes**

L. De Lathauwer, B. De Moor, and J. Vandewalle.
A multilinear singular value decomposition.
SIAM J. Matrix Anal. Appl., 21(4):1253–1278, 2000.

Ivan V Oseledets and Eugene E Tyrtyshnikov.
Breaking the curse of dimensionality, or how to use svd in many dimensions.
SIAM Journal on Scientific Computing, 31(5):3744–3759, 2009.

L. Grasedyck.
Hierarchical singular value decomposition of tensors.
SIAM J. Matrix Anal. Appl., 31:2029–2054, 2010.

M. Che and Y. Wei.
Randomized algorithms for the approximations of tucker and the tensor train decompositions.
Advances in Computational Mathematics, 45(1):395–428, 2019.

Y. Sun, Y. Guo, C. Luo, J. Tropp, and M. Udell.
Low-rank tucker approximation of a tensor from streaming data.
SIAM Journal on Mathematics of Data Science, 2(4):1123–1150, 2020.

B. Huber, R. Schneider, and S. Wolf.
A randomized tensor train singular value decomposition.
In *Compressed sensing and its applications*, pages 261–290. Springer, 2017.

V. Ehrlacher, L. Grigori, D. Lombardi, and H. Song.
Adaptive hierarchical subtensor partitioning for tensor compression.
SIAM Journal on Scientific Computing, 43(1):A139–A163, 2021.

L. Grigori and S. Kumar.
Parallel tensor train through hierarchical decomposition.
2020.

H. A. Daas, G. Ballard, and P. Benner.
Parallel algorithms for tensor train arithmetic.
arXiv preprint arXiv:2011.06532, 2020.

- **Relaxation methods**

M. Yuan and C.-H. Zhang.
On tensor completion via nuclear norm minimization.
Foundations of Computational Mathematics, 16(4):1031–1068, 2016.

H. Rauhut, R. Schneider, and Z. Stojanac.
Tensor Completion in Hierarchical Tensor Representations, pages 419–450.
Springer International Publishing, Cham, 2015.

H. Rauhut, R. Schneider, and Z. Stojanac.
Low rank tensor recovery via iterative hard thresholding.
Linear Algebra and its Applications, 523:220–262, 2017.

M. Bachmayr and R. Schneider.
Iterative methods based on soft thresholding of hierarchical tensors.
Foundations of Computational Mathematics, pages 1–47, 2016.

- **Approximation with structured sampling**

J. Ballani, L. Grasedyck, and M. Kluge.
Black box approximation of tensors in hierarchical tucker format.
Linear Algebra and its Applications, 438(2):639 – 657, 2013.
Tensors and Multilinear Algebra.

I. Oseledets and E. Tyrtyshnikov.
TT-cross approximation for multidimensional arrays.
Linear Algebra And Its Applications, 432(1):70–88, JAN 1 2010.

A. Nouy.
Higher-order principal component analysis for the approximation of tensors in tree-based low-rank formats.
Numerische Mathematik, 141(3):743–789, Mar 2019.

C. Haberstich, A. Nouy, and G. Perrin.
Active learning of tree tensor networks using optimal least-squares.
arXiv preprint arXiv:2104.13436, 2021.

- **Low-rank methods for tensor-structured equations: surveys**

B. B. Khoromskij and C. Schwab.
Tensor-structured Galerkin approximation of parametric and stochastic elliptic pdes.
SIAM Journal on Scientific Computing, 33(1):364–385, 2011.

B. Khoromskij.
Tensors-structured numerical methods in scientific computing: Survey on recent advances.
Chemometrics and Intelligent Laboratory Systems, 110(1):1 – 19, 2012.

A. Nouy.
Low-rank methods for high-dimensional approximation and model order reduction.
ArXiv e-prints, November 2015.

Markus Bachmayr, Reinhold Schneider, and André Uschmajew.
Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations.
Foundations of Computational Mathematics, pages 1–50, 2016.

- **Low-rank methods for tensor-structured equations: truncated iterations**

D. Kressner and C. Tobler.
Low-rank tensor Krylov subspace methods for parametrized linear systems.
SIAM Journal on Matrix Analysis and Applications, 32(4):1288–1316, 2011.

J. Ballani and L. Grasedyck.
A projection method to solve linear systems in tensor format.
Numerical Linear Algebra with Applications, 20(1):27–43, 2013.

M. Bachmayr and W. Dahmen.
Adaptive near-optimal rank tensor approximation for high-dimensional operator equations.
Foundations of Computational Mathematics, 15(4):839–898, 2015.

Markus Bachmayr and Reinhold Schneider.
Iterative methods based on soft thresholding of hierarchical tensors.
Foundations of Computational Mathematics, pages 1–47, 2016.

- **Alternating minimization and rank adaptation**

S. Holtz, T. Rohwedder, and R. Schneider.
The alternating linear scheme for tensor optimization in the tensor train format.
SIAM Journal on Scientific Computing, 34(2):A683–A713, 2012.

S. V. Dolgov and D. V. Savostyanov.
Alternating minimal energy methods for linear systems in higher dimensions.
SIAM Journal on Scientific Computing, 36(5):A2248–A2271, 2014.

- **Greedy algorithms**

E. Cances, V. Ehrlacher, and T. Lelievre.
Convergence of a greedy algorithm for high-dimensional convex nonlinear problems.
Mathematical Models & Methods In Applied Sciences, 21(12):2433–2467, December 2011.

A. Falcó and A. Nouy.
Proper generalized decomposition for nonlinear convex problems in tensor banach spaces.
Numerische Mathematik, 121:503–530, 2012.

- **Preconditioners in low-rank formats**

B. Khoromskij.
Tensor-structured preconditioners and approximate inverse of elliptic operators in $\mathbb{R}^d$.
Constructive Approximation, 30(3):599–620, 2009.

L. Giraldi, A. Nouy, and G. Legrain.
Low-rank approximate inverse for preconditioning tensor-structured linear systems.
SIAM Journal on Scientific Computing, 36(4):A1850–A1870, 2014.

M. Bachmayr and V. Kazeev.
Stability of low-rank tensor representations and structured multilevel preconditioning for elliptic pdes.
Foundations of Computational Mathematics, 20(5):1175–1236, 2020.

- **Learning with tensor networks**

L. Grasedyck and S. Krämer.
Stable ALS approximation in the TT-format for rank-adaptive tensor completion.
Numerische Mathematik, 143(4):855–904, 2019.

E. Stoudenmire and D. J. Schwab.
Supervised learning with tensor networks.
In *Advances in Neural Information Processing Systems*, pages 4799–4807, 2016.

E. Grelier, A. Nouy, and M. Chevreuil.
Learning with tree-based tensor formats.
arXiv e-prints, page arXiv:1811.04455, Nov. 2018.

E. Grelier, A. Nouy, and R. Lebrun.
Learning high-dimensional probability distributions using tree tensor networks.
arXiv preprint arXiv:1912.07913, 2019.

B. Michel and A. Nouy.
Learning with tree tensor networks: complexity estimates and model selection. To appear in Bernoulli.
arXiv e-prints, page arXiv:2007.01165, July 2020.

- **Geometry of tensor manifolds and geometrical approaches**

P-A Absil, Robert Mahony, and Rodolphe Sepulchre.
Optimization algorithms on matrix manifolds.
Princeton University Press, 2009.

S. Holtz, T. Rohwedder, and R. Schneider.
On manifolds of tensors of fixed TT-rank.
Numerische Mathematik, 120(4):701–731, 2012.

A. Uschmajew and B. Vandereycken.
Geometric Methods on Low-Rank Matrix and Tensor Manifolds, pages 261–313.
Springer International Publishing, Cham, 2020.

M. Billaud-Friess, A. Falco, and A. Nouy.
Principal bundle structure of matrix manifolds.
ArXiv e-prints, May 2017.

A. Falcó, W. Hackbusch, and A. Nouy.
On the Dirac–Frenkel variational principle on tensor Banach spaces.
Foundations of Computational Mathematics, 19(1):159–204, Feb 2019.

A. Falco, W. Hackbusch, and A. Nouy.
Geometry of tree-based tensor formats in tensor Banach spaces, 2021.

- **Dynamical low-rank methods for evolution problems**

C. Lubich, T. Rohwedder, R. Schneider, and B. Vandereycken.
Dynamical approximation by hierarchical tucker and tensor-train tensors.
SIAM Journal on Matrix Analysis and Applications, 34(2):470–494, 2013.

E. Musharbash.
Dynamical low rank approximation of pdes with random parameters.
page 194, 2017.

M. Bachmayr, H. Eisenmann, E. Kieri, and A. Uschmajew.
Existence of dynamical low-rank approximations to parabolic problems, 2020.

M. Billaud-Friess, A. Falcó, and A. Nouy.
A new splitting algorithm for dynamical low-rank approximation motivated by the fibre bundle structure of matrix manifolds, to appear in BIT Numerical Mathematics 2021.