Approximation and learning with tree tensor networks

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Joint work with
Mazen Ali (Approximation), Bertrand Michel (Learning)

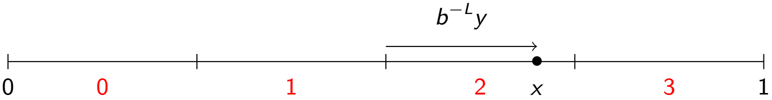
Outline

Tensorization of functions

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

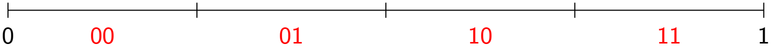
- For $b, L \in \mathbb{N}$, we subdivide uniformly the interval $[0,1)$ into b^L intervals. Any $x \in [0,1)$ can be written

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



- 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 multivariate functions

A function $f(x_1, \dots, x_d)$ 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

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

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

This defines a linear isometry (tensorization map)

$$T_{b,L} : L^p([0,1)^d) \to L^p(\{0, \dots, b-1\}^{Ld} \times [0,1)^d)$$

Feature tensor space

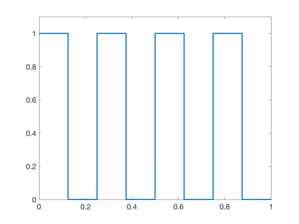
We consider functions whose tensorization at resolution L are in the tensor space (feature space)

$$V_L = (\mathbb{R}^b)^{\otimes Ld} \otimes S^{\otimes d}$$

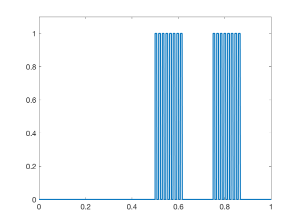
with S some subspace of univariate functions.

If $S = \mathbb{P}_k$, V_L is a space of multivariate splines of degree k over a uniform partition with b^{dL} elements.

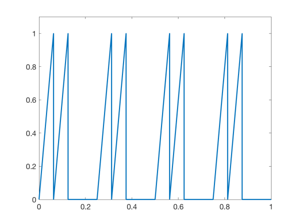
Examples of elementary tensors $f(x) = v^1(i_1)...v^L(i_L)v^{L+1}(y)$ $(b = 2)$



(a) $\delta_0(i_3)$



(b) $\delta_1(i_1)\delta_0(i_3)\delta_0(i_7)$



(c) $\delta_0(i_3)y$ $(L = 4)$

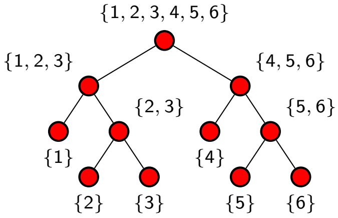
Tree based tensor format

We consider functions whose tensorization $\boldsymbol{f}$ in V_L is in a tree-based (hierarchical) tensor format $\mathcal{T}_r^T(V_L)$, which is a set of rank-structured tensors associated with a dimension tree T over $\{1, \dots, Ld + d\}$ and associated ranks $r = (r_\alpha)_{\alpha \in T}$.

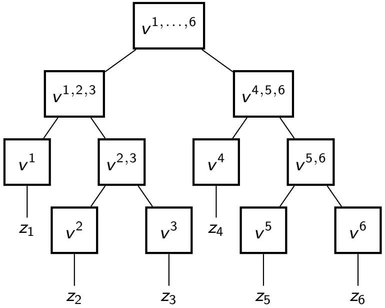
A tensor $\boldsymbol{f}$ in $\mathcal{T}_r^T(V_L)$ admits a multilinear parametrization

$$\boldsymbol{f}(z) = \sum_{\substack{1 \leq k_\gamma \leq r_\gamma \\ \gamma \in T}} \prod_{\alpha \in T \setminus \mathcal{L}(T)} v^\alpha_{k_\alpha, (k_\beta)_{\beta \in S(\alpha)}} \prod_{\alpha \in \mathcal{L}(T)} v^\alpha_{k_\alpha}(z_\alpha)$$

with a collection of parameters $\boldsymbol{v} = \{v^\alpha\}_{\alpha \in T}$ forming a tree tensor network.



Dimension tree T



Tree tensor network

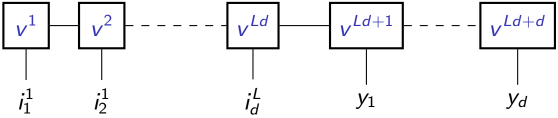
Tensor train format

For a linear tree

$$T_L = \{\{1\}, \{1,2\}, \ldots, \{1, \ldots, Ld + d\}\},$$

the set $\mathcal{T}_r^{T_L}(V_L)$ corresponds to the tensor train format and the corresponding function $f(x_1, \ldots, x_d)$ is in the Quantized Tensor Train (QTT) format [Kazeev, Khoromskij, Oseledets, Schwab, ...]

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



Approximation tools based on tree tensor networks

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 $f \in \Phi_{L,T_L,r}$ 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

$$\Phi_n^{\mathcal{F}} \subset \Phi_n^{\mathcal{S}} \subset \Phi_{a+bn^2}^{\mathcal{F}}$$

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)

Outline

Properties of tree tensor networks

Approximation tools based on tensor networks (with or without sparsity) 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

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

Direct embeddings of smoothness classes

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

Theorem (Continuous embedding of Besov spaces $B^\alpha_q(L^p)$)

For $\alpha > 0$ and $0 < p \leq \infty$, and an arbitrary $\tilde\alpha < \alpha$,

$$B^\alpha_\infty(L^p) \hookrightarrow \mathcal{F}^{\tilde\alpha/d}_\infty(L^p)$$

- Tensor networks achieve 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.

Direct embeddings of smoothness classes

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 (Continuous embedding of Besov spaces $B^\alpha_q(L^\tau)$)

Let $\alpha > 0$, $0 < \tau < p < \infty$ *and*

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

For any $\tilde\alpha < \alpha$ *and* $0 < q \leq \tau$,

$$B^\alpha_q(L^\tau) \hookrightarrow \mathcal{S}^{\tilde\alpha/d}_\infty(L^p) \hookrightarrow \mathcal{F}^{\tilde\alpha/(2d)}_\infty(L^p)$$

- 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 $O(n^{-\alpha/(2d)})$.

Direct embeddings of smoothness classes

- For Besov spaces with mixed dominating smoothness $MB^{\alpha}_{q}(L^p)$, sparse tensor networks achieve near to optimal performance in $O(n^{-\alpha}\log(n)^d)$.
- For Besov spaces with anisotropic smoothness $AB^{\boldsymbol{\alpha}}_{q}(L^p)$, sparse tensor networks also achieve near to optimal approximation rates in $n^{-s(\boldsymbol{\alpha})/d}$ with

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

  the aggregated 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.

Outline

Statistical learning

Two typical tasks of statistical learning are to

- approximate a random variable Y by a function of random variables $X = (X_1, \ldots, X_d)$, from samples of the pair $Z = (X, Y)$ (supervised learning)
- approximate the probability distribution of a random variable $X = (X_1, \ldots, X_d)$ from samples of the distribution (unsupervised learning)

Statistical learning

A classical approach is to introduce a risk functional $\mathcal{R}(f)$ whose minimizer over the set of functions f is the target function $f^\star$, and such that the excess risk

$$\mathcal{E}(f) := \mathcal{R}(f) - \mathcal{R}(f^\star)$$

measures some distance between the target $f^\star$ and the function f.

The risk is defined as an expectation

$$\mathcal{R}(f) = \mathbb{E}(\gamma(f, Z))$$

where γ is a contrast (or loss) function, and $Z = X$ or (X, Y).

- For least-squares regression in supervised learning, $\mathcal{R}(f) = \mathbb{E}((Y - f(X))^2)$, $f^\star(X) = \mathbb{E}(Y|X)$ and

$$\mathcal{E}(f) = \|f^\star - f\|^2_{L^2_\mu}, \quad \text{with} \quad X \sim \mu$$

- For density estimation with L^2-loss, $\mathcal{R}(f) = \mathbb{E}(\|f\|^2_{L^2_\mu} - 2f(Z))$ and

$$\mathcal{E}(f) = \|f^\star - f\|^2_{L^2_\mu}$$

  is the L^2 distance between f and the probability density $f^\star$ of X with respect to a reference measure μ.

Empirical risk minimization

Given i.i.d. samples $z_1, \dots, z_N$ of Z, an approximation $\hat{f}_M$ of $f^\star$ can be obtained by minimizing the empirical risk

$$\widehat{\mathcal{R}}_N(f) = \frac{1}{N} \sum_{i=1}^{N} \gamma(f, z_i)$$

over a certain model class M.

- The error (excess risk)

$$\mathcal{E}(\hat{f}_M) = \underbrace{\inf_{f \in M} \mathcal{R}(f) - \mathcal{R}(f^\star)}_{\text{approximation error}} + \underbrace{\mathcal{R}(\hat{f}_M) - \inf_{f \in M} \mathcal{R}(f)}_{\text{estimation error}}$$

- For a given sample, when taking larger and larger model classes, approximation error $\searrow$ while estimation error $\nearrow$.
- Methods should be proposed for the selection of a model class taking the best from the available information.

Model selection for tree tensor networks

Consider the approximation tool Φ made of a countable collection of tree tensor networks $(M_m)_{m \in \mathcal{M}}$ associated with different resolutions L_m, trees T_m, ranks r_m, and sparsity patterns Λ_m.

We use a model selection approach à la Barron, Birgé and Massart, which consists in selecting the model $\hat{m}$ minimizing a penalized empirical risk

$$\min_{m \in \mathcal{M}} \widehat{\mathcal{R}}_N(\hat{f}_m) + pen(m)$$

with

- $\hat{f}_m$ a minimizer of the empirical risk $\widehat{\mathcal{R}}_N(f)$ over M_m,
- $pen(m)$ a penalty depending on the complexity C_m of the model class M_m.

Model selection for tree tensor networks

A first step is to obtain an upper bound of the metric entropy of the set of tensor networks M_m with bounded parameters

$$\log \mathcal{N}(\epsilon, M_m, \|\cdot\|_{L^\infty}) \leq C_m \log(6\epsilon^{-1} B |T_L|)$$

Then in a least-squares setting, by adapting a result of Koltchinskii, we prove that the selected estimator $\hat{f}_{\hat{m}}$ satisfies an oracle inequality

$$\mathbb{E}(\mathcal{E}(\hat{f}_{\hat{m}})) \lesssim \inf_{m \in \mathcal{M}} \left(\inf_{f \in M_m} \mathcal{E}(f) + \kappa pen(m) \right) + \frac{\kappa'}{N}$$

if the penalty is taken as

$$pen(m) \sim C_m \log(C_m) \frac{\log(N)}{N} + \frac{\log(\mathcal{N}_{C_m}(\mathcal{M}))}{N}$$

with $\mathcal{N}_c(\mathcal{M})$ the number of models with complexity c,

$$\mathcal{N}_c(\mathcal{M}) = |\{m \in \mathcal{M} : C_m = c\}|.$$

Model selection for tree tensor networks

For the collection $\mathcal{M}$ of all possible tensor networks (full or sparse, with or without variable trees),

$$\log(\mathcal{N}_c(\mathcal{M})) \lesssim c \log(c),$$

and the estimator $\hat{f}_{\hat{m}}$ selected by our procedure satisfies

$$\mathbb{E}(\mathcal{E}(\hat{f}_{\hat{m}})) \lesssim \inf_{m \in \mathcal{M}} \inf_{f \in M_m} \mathcal{E}(f) + C_m \log(C_m) \frac{\log(N)}{N}$$

For a target function $f^\star$ in the approximation class $\mathcal{A}^\alpha_\infty(\Phi, L^2)$, we have

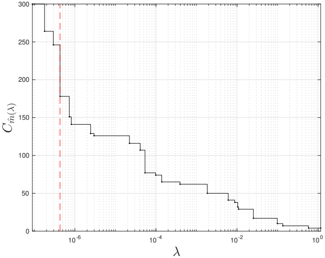
$$\mathbb{E}(\|\hat{f}_{\hat{m}} - f^\star\|^2_{L^2}) \lesssim N^{-\frac{2\alpha}{2\alpha+1}} \log(N)^{\frac{4\alpha}{2\alpha+1}}$$

Miminax rates for Besov spaces

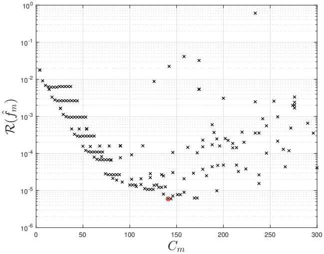
- Minimax rates in $O(N^{-\frac{2s}{2s+d}})$ are known to be achieved with linear estimators for $B^s_q(L^2)$ or nonlinear estimators for $B^s_q(L^\tau)$, $\tau < 2$.
- Arbitrarily close to minimax rates in $O(N^{-\frac{2\tilde{s}}{2\tilde{s}+d}})$ ($\tilde{s} < s$) achieved by full tensors networks for $B^s_q(L^2)$ and sparse tensor networks for $B^s_q(L^\tau)$.
- The proposed strategy is (near to) minimax adaptive to a wide range of Besov classes, without requiring to adapt the tool to the regularity of the target.
- Rates arbitrarily close to minimax also achieved for target functions with mixed dominating smoothness or anisotropic smoothness, using sparse tensor networks. Slightly worse rates for full tensor networks.

From theory to practice

- The penalty is chosen as $pen(m) = \lambda \frac{C_m}{N}$ and the parameter λ is estimated using slope heuristics.



(d) Function $\lambda \mapsto C_{\hat{m}(\lambda)}$, λ^{cj} (red).



(e) Points $(C_m, \mathcal{R}(\hat{f}_m))$, $m \in \mathcal{M}$, and selected model (red).

- We consider collections of models with variable trees T_m and ranks r_m. An exhaustive exploration of possible ranks and trees is impossible (combinatorial complexity).

  In practice, we generate a collection of models using an adaptive algorithm with suitable exploration strategy [Grelier, Nouy and Chevreuil 2018, Michel and Nouy 2020].

From theory to practice

- The obtained theoretical results requires minimizers $\hat{f}_m$ of the empirical risk over model classes M_m. Although adaptive algorithms perform well in practice, there is no guarantee to find a solution of

$$\min_{f \in M_m} \widehat{\mathcal{R}}_N(f)$$

About the tree or architecture of the network

For classical smoothness classes, optimal approximation rates and minimax results are obtained with fixed linear trees T_L (tensor train format) or any other fixed binary tree with a suitable ordering of the variables.

But in practice, a much better performance can be observed when adapting the tree to the target function, i.e. considering an approximation tool with free tree

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

- Higher nonlinearity but how much higher ?
- Higher power but how much higher ?

References and Software

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

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

M. Ali and A. Nouy.
Approximation with tensor networks. part iii: Multivariate approximation.
ArXiv, arXiv:2101.11932, 2021.

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

E. Grelier, A. Nouy, M. Chevreuil.
Learning with tree-based tensor formats.
Arxiv eprints, arXiv:1811.04455, Nov. 2018.

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).