Approximation and learning with tree tensor networks

Anthony Nouy

Centrale Nantes
Laboratoire de Mathématiques Jean Leray

Collaborators:
Antonio Falco, Wolfgang Hackbusch (topology and geometry of tensor networks),
Mazen Ali, Markus Bachmayr, Reinhold Schneider (approximation power),
Bertrand Michel (complexity estimates and model selection),
Mathilde Chevreuil, Loic Giraldi, Erwan Grelier, Régis Lebrun (learning algorithms),
Cécile Haberstich, Guillaume Perrin (active learning algorithms)

Approximation/Learning in high dimension

Many problems in statistics and machine learning require the approximation of functions of many variables

$$u(x_1, \dots, x_d)$$

- **Supervised learning**.
  Approximation of a random variable Y by a function of a set of random variables $X = (X_1, \dots, X_d)$, using samples of (X, Y). The approximation is used as a predictive model.

- **Unsupervised learning**.
  Estimation of the probability distribution of a random vector $X = (X_1, \dots, X_d)$, from samples of X.

These are two typical tasks in **uncertainty quantification**, where Y is some output variable of a (numerical or experimental) model depending on a set of random parameters X.

Approximation/Learning in high dimension

The function u is approximated by an element of an approximation set F_n described by n parameters, also called model class or hypothesis set.

A sequence $(F_n)_{n\geq 1}$ of approximation sets is called an approximation tool. Standard approximation tools include splines, wavelets, polynomials, kernel functions.

A fundamental question is to determine the complexity $n = n(\epsilon, u)$ to obtain an approximation error ϵ.

For a function u from classical regularity classes (Sobolev, Besov or even analytic functions), the complexity $n(\epsilon, u)$ typically grows exponentially with d for any "reasonable" approximation tool, that is the curse of dimensionality.

How to beat the curse of dimensionality ?

We have to

- make further assumptions on the function, going ahead classical regularity assumptions,
- and propose ad-hoc approximation tools.

We would like

- an approximation tool that achieves a good performance for many classes of functions,
- algorithms that practically compute approximations achieving a certain precision with near optimal complexity.

A good candidate is provided by tree tensor networks, that are related to low-rank structures of multivariate functions.

Outline

Outline

Ranks of multivariate functions

Consider a multivariate function

$$v(x_1, \dots, x_d)$$

For a subset of variables

$$\alpha \subset \{1, \dots, d\} := D,$$

the function v can be identified with a bivariate function

$$v(x_\alpha, x_{\alpha^c})$$

where x_α and x_{α^c} are complementary groups of variables.

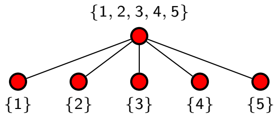
The rank of this bivariate function is called the α-rank of v, denoted $\mathrm{rank}_\alpha(v)$. It is the minimal integer r_α such that

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

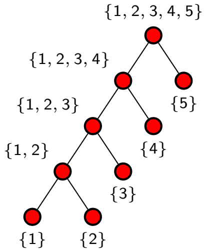
Low rank tensor formats

- For T a collection of subsets of D and a given tuple $r = (r_\alpha)_{\alpha \in T}$, a tensor format is defined by
$$\mathcal{T}_r^T(\mathcal{H}) = \bigcap_{\alpha \in T} \{v \in \mathcal{H} : \mathrm{rank}_\alpha(v) \leq r_\alpha\}$$
where $\mathcal{H}$ is some finite dimensional space of multivariate functions.

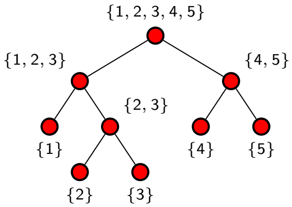
- In the particular case where T is a dimension partition tree, $\mathcal{T}_r^T(\mathcal{H})$ is a tree-based tensor format.



Tucker format



Tensor Train format



Hierarchical Tucker format

Tree-based tensor formats as tree tensor networks

Consider a tensor space $\mathcal{H} = \mathcal{H}_1 \otimes \ldots \otimes \mathcal{H}_d$ of multivariate functions, and let $\{\phi^\nu_{i_\nu} : 1 \le i_\nu \le N_\nu\}$ be a basis of $\mathcal{H}_\nu$, e.g. splines, wavelets, or a set of features.

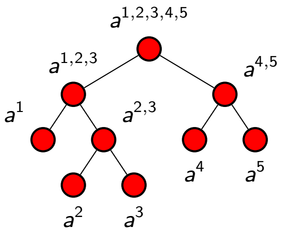
A function $v \in \mathcal{T}^T_r(\mathcal{H})$ admits a representation

$$v(x) = \sum_{1\le i_1 \le N_1} \ldots \sum_{1 \le i_d \le N_d} a_{i_1,\ldots,i_d} \phi^1_{i_1}(x_1) \ldots \phi^d_{i_d}(x_d)$$

with a tensor $a \in \mathbb{R}^{N_1 \times \ldots \times N_d}$ having an explicit representation

$$a_{i_1,\ldots,i_d} = \sum_{\substack{1 \le k_\beta \le r_\beta \\ \beta \in T}} \prod_{\alpha \in \mathcal{I}(T)} a^\alpha_{(k_\beta)_{\beta\in S(\alpha)},k_\alpha} \prod_{\nu=1}^d a^\nu_{i_\nu,k_\nu}$$

with parameters $\{a^\alpha\}_{\alpha\in T}$ forming a tree tensor network.

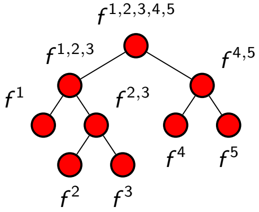


Tree-based tensor formats as compositional functions

By identifying a tensor $a^\alpha \in \mathbb{R}^{n_1 \times \ldots \times n_s \times r_\alpha}$ 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 $\mathcal{T}_r^T(\mathcal{H})$ admits a representation as a tree-structured composition of multilinear functions $\{f^\alpha\}_{\alpha \in T}$.



For the above tree,

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

where $\Phi^\nu(x_\nu) = (\phi^\nu_{i_\nu}(x_\nu))_{1 \le i_\nu \le N_\nu}$ is a vector of N_ν features in the variable x_ν.

Tree-based tensor format as a deep neural network

It corresponds to a deep neural network with

- a sum-product architecture (multilinear units)
- a depth L bounded by $d-1$,
- a width at level ℓ related to the α-ranks of the nodes α of level ℓ.
- a sparse connectivity given by T

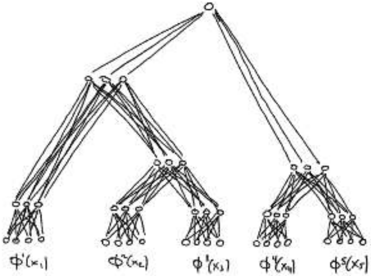


Figure: Number of features per variable $N_\nu = 4$ and ranks $r_\alpha = 3$.

- E.g. convolutional network for a balanced tree, recurrent network for a linear tree

Representation complexity of tree tensor networks

The number of parameters (representation complexity) is

$$C(T, r, \mathcal{H}) = \sum_{\alpha \in \mathcal{I}(T)} r_\alpha \prod_{\beta \in S(\alpha)} r_\beta + \sum_{\nu=1}^{d} r_\nu N_\nu .$$

Noting that $\# T \leq 2d - 1$,

$$C(T, r, \mathcal{H}) \leq R^s + (d-2) R^{s+1} + dRN,$$

where s is the arity of the tree, R the maximum rank, and N the maximum number of features per variable.

Outline

Approximation power of tree tensor networks

We want to quantify the best approximation error

$$e_r^T(u) = \min_{v \in \mathcal{T}_r^T(\mathcal{H})} \|u - v\|$$

for a target function u in some function classes and compare it with other approximation tools.

We consider functions in $L^2_\mu(\mathcal{X})$ where $\mathcal{X} = \mathcal{X}_1 \ldots \times \mathcal{X}_d$ is equipped with a product measure μ, and

$$\|u - v\|^2 = \int_{\mathcal{X}} (u(x) - v(x))^2 d\mu(x).$$

Linear widths of multivariate functions

For each $\alpha \in T$, we start by considering the approximation error by functions with bounded α-rank,

$$e^{\alpha}_{r_\alpha}(u)^2_{L^2} = \inf_{\text{rank}_\alpha(v) \leq r_\alpha} \|u - v\|^2 = \inf_{\text{rank}_\alpha(v) \leq r_\alpha} \int_{\mathcal{X}_{\alpha^c}} \|u(\cdot, x_{\alpha^c}) - v(\cdot, x_{\alpha^c})\|^2_{L^2_{\mu_\alpha}} d\mu_{\alpha^c}(x_{\alpha^c}),$$

which is equivalent to

$$e^{\alpha}_{r_\alpha}(u)^2_{L^2} = \inf_{\dim(V_r) = r} \int_{\mathcal{X}_{\alpha^c}} \|u(\cdot, x_{\alpha^c}) - \mathcal{P}_{V_r} u(\cdot, x_{\alpha^c})\|^2_{L^2_{\mu_\alpha}} d\mu_{\alpha^c}(x_{\alpha^c})$$

where the infimum is taken over all r-dimensional subspaces.

By considering the set of partial evaluations of u

$$K_\alpha(u) = \{u(\cdot, x_{\alpha^c}) : x_{\alpha^c} \in \mathcal{X}_{\alpha^c}\} \subset L^2_{\mu_\alpha}(\mathcal{X}_\alpha),$$

we have

$$e^{\alpha}_{r}(u)_{L^2} \leq \inf_{\dim(V_r) = r} \sup_{f \in K_\alpha(u)} \|f - \mathcal{P}_{V_r} f\| := d_r(K_\alpha(u))_{L^2_{\mu_\alpha}},$$

the upper bound being the Kolmogorov r-width of $K_\alpha(u)$.

Approximation power of tree tensor networks

For a given tree, given a collection of spaces U_α with dimension r_α, $\alpha \in T \setminus \{D\}$, the approximation

$$u_r = \prod_{\alpha \in T} \mathcal{P}_{U_\alpha} u$$

obtained by successive orthogonal projections (suitably ordered) satisfies $u_r \in \mathcal{T}_r^T(U)$, with $U = U_1 \otimes \ldots \otimes U_d$ and

$$\|u - u_r\|^2 \leq \sum_{\alpha \in T \setminus \{D\}} \|u - \mathcal{P}_{U_\alpha} u\|^2$$

Taking the infimum over the spaces U_α for interior nodes $\alpha \in \mathcal{I}(T)$ and taking $U_\nu = \mathcal{H}_\nu$ for leaf nodes ν, we deduce that the best approximation error in $\mathcal{T}_r^T(\mathcal{H})$ satisfies

$$e_r^T(u)_{L^2}^2 \leq \sum_{\alpha \in \mathcal{I}(T) \setminus \{D\}} e_{r_\alpha}^\alpha(u)^2 + \sum_{\nu=1}^{d} \|u - \mathcal{P}_{\mathcal{H}_\nu} u\|^2$$

Then error bounds can be obtained with information on the linear widths of the sets $K_\alpha(u)$ of partial evaluations of u.

Approximation power for Sobolev classes

- For functions $u \in H^s((0,1)^d)$, partial evaluations also have Sobolev regularity, and from results on Kolmogorov widths of Sobolev balls, we deduce that
$$e^{\alpha}_{r_\alpha}(u) \lesssim {r_\alpha}^{-s/d_\alpha} \|u\|_{H^s}, \quad d_\alpha = \min\{\#\alpha, d - \#\alpha\}.$$
Using suitable spaces $\mathcal{H}_\nu$ (e.g. splines), we obtain that the complexity to achieve a precision ϵ (whatever the tree) is
$$n(\epsilon, u) \lesssim \epsilon^{-d/s},$$
which is the ideal performance we can expect, which is achieved by other tools (e.g. multivariate splines).

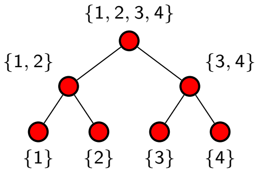
- For functions u in mixed Sobolev classes $H^s_{mix}(\mathbb{T}^d)$, from bounds on Kolmogorov widths of mixed Sobolev balls, we obtain
$$e^{\alpha}_{r_\alpha}(u) \lesssim r_\alpha^{-s} \log(r_\alpha)^{s(d_\alpha - 1)}$$
and a complexity to achieve a precision ϵ (with binary trees)
$$n(\epsilon, u) \lesssim \epsilon^{-3/s} \log(\epsilon^{-1})^d$$
almost the ideal performance achieved by hyperbolic cross (sparse) approximation.

- Other bounds in [Schneider and Uschmajew 2014] using results on bilinear approximation [Temlyakov 1989].

Approximation power for compositional functions [with Markus Bachmayr and Reinhold Schneider]

Tree tensor networks they can perform much better for non standard classes of functions, e.g. a tree-structured composition of regular functions $\{u_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$u(x) = u_{1,2,3,4}\left(u_{1,2}\left(u_1(x_1), u_2(x_2)\right), u_{3,4}\left(u_3(x_3), u_4(x_4)\right)\right)$$



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

$$n(\epsilon, u) \lesssim \epsilon^{-3/s} L^3 B^{3L} d^{1+3/2s}$$

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

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

Deep versus shallow networks

- A function in canonical format (shallow network)

$$u(x) = \sum_{k=1}^{r} u_k^1(x_1)\dots u_k^d(x_d)$$

can be represented in tree-based format with a similar complexity.

- Conversely, a typical function in tree-based format $\mathcal{T}_r^T$ has a canonical rank depending exponentially in d.

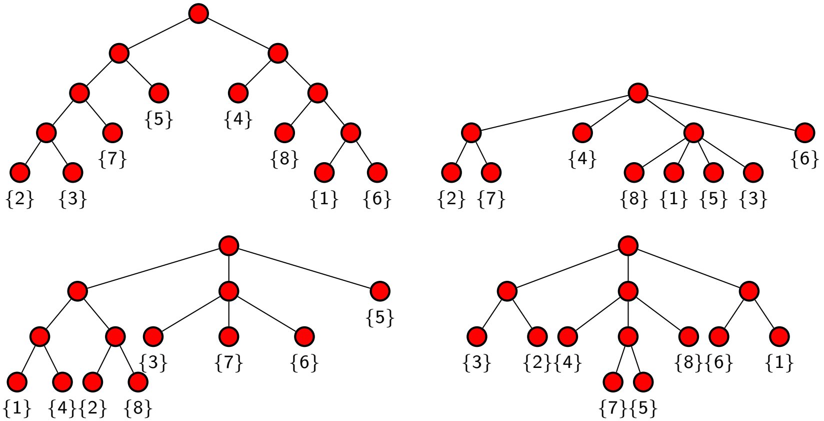
Deep is better !

For a balanced or linear binary tree T, the subset of tensors v in $\mathcal{T}_r^T(\mathbb{R}^{n\times\dots\times n})$ with canonical rank less than $\min\{n, r\}^{d/2}$ is of Lebesgue measure 0 [Cohen et al. 2016, Khrulkov et al 2018]

- But a typical function in $\mathcal{T}_r^T$ may admit a representation complexity exponential in d when using another tree.

Selection of a tree

Choosing a good tree (architecture of network) is a crucial but combinatorial problem...



Stochastic algorithms for tree optimization (with some heuristics) proposed in [Grelier, Nouy and Chevreuil 2018].

Outline

Risk

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

$$\mathcal{R}(v) - \mathcal{R}(u)$$

measures some distance between the target u and the function v.

The risk is defined as an expectation

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

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

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

$$\mathcal{R}(v) - \mathcal{R}(u) = \mathbb{E}((u(X) - v(X))^2) = \|u - v\|^2_{L^2_\mu}$$

  with $X \sim \mu$.

- For density estimation with L^2-loss, $\mathcal{R}(v) = \mathbb{E}(\|v\|^2_{L^2_\mu} - 2v(Z))$ and $\mathcal{R}(v) - \mathcal{R}(u) = \|u - v\|^2_{L^2_\mu}$ is the L^2 distance between v and the probability density u of Z with respect to a reference measure μ.

Empirical risk minimization

Given i.i.d. samples $\{z_i\}_{i=1}^n$ of Z, an approximation $\hat{u}_F^n$ of u can be obtained by minimization of the empirical risk

$$\widehat{\mathcal{R}}_n(v) = \frac{1}{n}\sum_{i=1}^n \gamma(v, z_i)$$

over a certain model class F.

- Denoting by u_F the minimizer of the risk over F, the error

$$\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u) = \underbrace{\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u_F)}_{\text{estimation error}} + \underbrace{\mathcal{R}(u_F) - \mathcal{R}(u)}_{\text{approximation error}}$$

- 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.

Estimation error

Given a model class F, a minimizer u_F of the risk $\mathcal{R}$ over F and a minimizer $\hat{u}_F^n$ of the empirical risk $\widehat{\mathcal{R}}_n$ over F, the estimation error

$$\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u_F) \le \mathcal{R}(\hat{u}_F^n) - \widehat{\mathcal{R}}_n(\hat{u}_F^n) + \widehat{\mathcal{R}}_n(u_F) - \mathcal{R}(u_F)$$

so that

$$\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u_F) \le 2 \sup_{v \in F} |\widehat{\mathcal{R}}_n(v) - \mathcal{R}(v)| = 2 \sup_{v \in F} |\frac{1}{n} \sum_{i=1}^{n} \gamma(v, z_i) - \mathbb{E}(\gamma(v, Z))|$$

and an error bound can be obtained by analyzing the fluctuations of $\frac{1}{n}\sum_{i=1}^{n} \gamma(v, z_i)$ around its mean.

Estimation error

Assume that F is compact in L^∞, and that for all $v, w \in F$, the contrast is uniformly bounded and Lipschitz,

$$|\gamma(v,Z)| \leq B, \quad |\gamma(v,Z) - \gamma(w,Z)| \leq L\|v-w\|_{L^\infty}$$

Then using a standard concentration inequality (here Hoeffding), we obtain

$$\mathbb{P}(\sup_{v\in F} |\frac{1}{n}\sum_{i=1}^n \gamma(v,z_i) - \mathbb{E}(\gamma(v,Z))| \geq 2\epsilon B) \leq 2\mathcal{N}_{\frac{\epsilon B}{2L}} e^{-\frac{n\epsilon^2}{2}}$$

where $\mathcal{N}_{\frac{\epsilon B}{2L}} = \mathcal{N}(\frac{\epsilon B}{2L}, F, \|\cdot\|_{L^\infty})$ is the covering number of F.

Metric entropy of tree tensor networks
[with Bertrand Michel]

Assume $\mathcal{H} \subset L^\infty(\mathcal{X})$ with basis functions $\{\phi_i\}_{i\in I}$ normalized in $L^\infty(\mathcal{X})$.

For any representation of $v \in \mathcal{T}_r^T(\mathcal{H})$ with parameters $\{f^\alpha : \alpha \in T\}$, the multilinearity of the parametrization implies

$$\|v\|_{L^\infty} \le \prod_\alpha \|f^\alpha\|_{\alpha,\infty},$$

with a suitable choice of norms

$$\|f^\alpha\|_{\alpha,\infty} = \sup_{\|z_\beta\|_\infty \le 1} \|f^\alpha((z_\beta)_{\beta\in S(\alpha)})\|_\infty$$

Considering the model class

$$F = RF_1, \quad F_1 = \{v \in \mathcal{T}_r^T(\mathcal{H}) : \max_{\alpha\in T} \|f^\alpha\|_{\alpha,\infty} \le 1\},$$

we obtain an upper bound of the metric entropy

$$\log \mathcal{N}(\epsilon, F, \|\cdot\|_{L^\infty}) \le C_F \log(6\epsilon^{-1} R \# T),$$

where $C_F = C(T, r, \mathcal{H})$ is the representation complexity of elements of F.

Coming back to estimation error

We conclude that if

$$n \geq 8\epsilon^{-2} B^2 \left(\log(2\eta^{-1}) + C_F \log(12\epsilon^{-1} RL \# T) \right)$$

then

$$\mathbb{P}(\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u_F) \leq \epsilon) \geq 1 - \eta.$$

Also

$$\mathbb{E}(\mathcal{R}(\hat{u}_F^n) - \mathcal{R}(u_F)) \lesssim B \sqrt{\frac{C_F}{n}} \sqrt{\log(n)}$$

Note that better bounds may be obtained using refined concentration inequalities for suprema of empirical processes

$$\sup_{v \in F} \left| \frac{1}{n} \sum_{i=1}^{n} \gamma(v, z_i) - \mathbb{E}(\gamma(v, Z)) \right|$$

Learning algorithm for tree tensor networks

A function v in the model class $\mathcal{T}_r^T(\mathcal{H})$ has a representation $v(x) = \Psi(x)((a^\alpha)_{\alpha \in T})$ where each parameter a^α is in a tensor space $\mathbb{R}^{K^\alpha}$ and $\Psi(x)$ is a multilinear map.

The empirical risk minimization problem over the nonlinear model class $\mathcal{T}_r^T$

$$\min_{(a^\alpha)_{\alpha \in T}} \frac{1}{n} \sum_{i=1}^{n} \gamma(\Psi(\cdot)((a^\alpha)_{\alpha \in T}), z_i)$$

can be solved using an alternating minimization algorithm, solving at each step an empirical risk minimization problem with a linear model

$$\Psi(x)((a^\alpha)_{\alpha \in T}) = \sum_{k \in K^\alpha} \Psi_k^\alpha(x) a_k^\alpha$$

with functions $\Psi_k^\alpha(x)$ depending on fixed parameters a^β, $\beta \neq \alpha$.

- In a L^2 setting, possible re-parametrization for having orthonormal functions $\Psi_k^\alpha(x)$.
- Sparsity in the tensors a^α can be exploited in different ways, e.g. by proposing different sparsity patterns and use a model selection technique (e.g. based on validation).
- For a leaf node ν, the approximation space $\mathcal{H}^\nu$ can be selected from a candidate sequence of spaces $\mathcal{H}_0^\nu \subset \ldots \subset \mathcal{H}_L^\nu \subset \ldots$

Learning algorithm for tree tensor networks

Selection an optimal model class $\mathcal{T}_r^T(\mathcal{H})$ is a combinatorial problem.

An algorithm is proposed in [Grelier, Nouy, Chevreuil 2018] that performs adaptations of the tree T (architecture), the rank r (widths) and the approximation space $\mathcal{H}$.

Start with an initial tree T and learn an approximation $v \in \mathcal{T}_r^T(\mathcal{H})$ with rank $r = (1, ..., 1)$. Then repeat

- Increase some ranks r_α based on estimates of truncation errors

$$\min_{\mathrm{rank}_\alpha(v) \le r_\alpha} \mathcal{R}(v) - \mathcal{R}(u)$$

- Learn an approximation v in $\mathcal{T}_r^T(\mathcal{H})$, with adaptive selection of $\mathcal{H}$
- Optimize the tree for reducing the storage complexity of v (stochastic algorithm using a suitable distribution over the set of trees)

$$\min_T C(T, \mathrm{rank}_T(v), \mathcal{H})$$

Model selection

Algorithms generate a sequence of estimations $\hat{u}_m^n$ in different model classes $(F_m)_{m\in\mathcal{M}}$ (with different trees T_m, ranks r_m and background approximation spaces $\mathcal{H}_m$).

The model selection approach of Barron, Birgé and Massart can be used, which consists in minimizing a penalized empirical risk

$$\min_{m\in\mathcal{M}} \widehat{\mathcal{R}}_n(\hat{u}_m^n) + \lambda pen(m)$$

with a penalty $pen(m)$ depending on the complexity C_{F_m} of the model.

This yields a model $\hat{m}$ that satisfies oracle inequalities.

In practice, the parameter λ can be estimated with slope heuristics.

Improving estimation error

- A better performance can be obtained by using as empirical risk

$$\widehat{\mathcal{R}}_n(v) = \frac{1}{n}\sum_{i=1}^{n} w_i \gamma(v, \tilde{z}_i)$$

  where the $\tilde{z}_i$ are i.i.d. samples from a measure $dP_{\tilde{Z}} = w(z)dP_Z$ and the weights $w_i = w(\tilde{z}_i)^{-1}$.

- The choice of sampling measure should be adapted to the risk and model class, and may be deduced from concentration inequalities.
  See [Cohen and Migliorati 2017][Haberstich, Nouy, Perrin 2020] for least-squares regression and linear models.

Improving estimation error [with C. Haberstich and G. Perrin]

- For tree tensor networks, use a specific sampling measure for each parameter, and estimate the parameters sequentially (from leaves to root).

  For a given α, the risk

$$\mathcal{R}(v) = \mathbb{E}((Y - v(X))^2) = \mathbb{E}(\mathbb{E}((Y - v(X))^2 | X_{\alpha^c}))$$

  is estimated by

$$\widehat{\mathcal{R}}_n(v) \approx \frac{1}{p} \sum_{j=1}^{p} \frac{1}{q} \sum_{i=1}^{q} (y^{i,j} - v(x_\alpha^i, x_{\alpha^c}^j))^2$$

  where samples $(x_\alpha^i, x_{\alpha^c}^j)$ are not independent, and the x_α^i are generated according a measure depending on the previously estimated tensors.

  Link to empirical principal component analysis of multivariate functions [Nouy 2019], for the estimation of optimal spaces U_α for the approximation of partial evaluations $u(\cdot, x_{\alpha^c})$.

Concluding remarks

- For classical sampling, the obtained theoretical results are related to the minimizer $\hat{u}_F^n$ of the empirical risk over the model class $F = \mathcal{T}_r^T(\mathcal{H})$, but available algorithms do not guarantee to find a solution of

$$\min_{v \in \mathcal{T}_r^T(\mathcal{H})} \widehat{\mathcal{R}}_n(v)$$

- Convexification of tree tensor networks ?

Available software

- Tensor formats and related algorithms are available in a Matlab toolbox ApproximationToolbox, available on GitHub.

  A. Nouy, E. Grelier, and L. Giraldi.
  ApproximationToolbox.
  February 2020. doi:10.5281/zenodo.3653971.

- A python package will be available soon.
- See https://anthony-nouy.github.io/software.html

Thank you for your attention

References

A. Nouy.
Low-rank methods for high-dimensional approximation and model order reduction.
In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, editors, *Model Reduction and Approximation: Theory and Algorithms*. SIAM, Philadelphia, PA, 2017.

A. Falcó, W. Hackbusch, and A. Nouy.
Tree-based tensor formats.
SeMA Journal, Oct 2018.

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

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.

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

C. Haberstich, A. Nouy, and G. Perrin.
Boosted optimal weighted least-squares.
arXiv e-prints, page arXiv:1912.07075, Dec 2019.

W. Hackbusch.
Tensor spaces and numerical tensor calculus, volume 42 of *Springer series in computational mathematics*.
Springer, Heidelberg, 2012.

A. Cohen and G. Migliorati.
Optimal weighted least-squares methods.
SMAI Journal of Computational Mathematics, 3:181–203, 2017.

N. Cohen, O. Sharir, and A. Shashua.
On the expressive power of deep learning: A tensor analysis.
In *Conference on Learning Theory*, pages 698–728, 2016.

Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets.
Expressive power of recurrent neural networks.
In *International Conference on Learning Representations*, 2018.

R. Schneider and A. Uschmajew.
Approximation rates for the hierarchical tensor format in periodic sobolev spaces.
Journal of Complexity, 30(2):56 – 71, 2014.
Dagstuhl 2012.

V. N. Temlyakov.
Bilinear approximation and applications.
Trudy Matematicheskogo Instituta imeni VA Steklova, 187:191–215, 1989.