Workshop "Apprentissage et simulation en grande dimension",
Airbus Group, June 24-26, 2019

Deep tensor networks

Part III: Supervised learning

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Supervised learning

We consider the problem of constructing an approximation of a random variable Y by a function of a set of random variables $X = (X_1, \ldots, X_d)$, using samples of (X, Y).

A first situation is when we are given independent samples of (X, Y) (passive learning).

Another situation is when the samples of (X, Y) can be generated from adaptively chosen samples of X (active learning) and the evaluation of a function u such that

$$Y = u(X) + \epsilon.$$

Outline

1. Passive learning through empirical risk minimization
   - Parametrization of tree-based formats
   - Rank adaptation
   - Tree adaptation

2. Active learning based on empirical principal component analysis
   - Principal component analysis of multivariate functions
   - Adaptive sampling based on principal component analysis

Outline

Empirical risk minimization

Consider a function v providing a prediction $v(x)$ for a given instance x of X.

Given a loss function ℓ such that $\ell(y, v(x))$ measures a certain dissimilarity between y and the prediction $v(x)$, we define the risk

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

whose minimizer $u(X)$ over the set of measurable functions (if it exists) is called the oracle.

Given i.i.d. samples $\{(x_i, y_i)\}_{i=1}^n$, an approximation $\hat{u}_F^n$ of u is obtained by minimization of the empirical risk

$$\widehat{\mathcal{R}}_n(v) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, v(x_i))$$

over a set of functions F (model class).

Least-squares regression

When choosing the square loss $\ell(y, z) = (y - z)^2$, the risk is

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

and the empirical risk

$$\mathcal{R}(v) = \frac{1}{n} \sum_{i=1} (y_i - v(x_i))^2.$$

The oracle is the conditional expectation $u(X) = \mathbb{E}(Y|X)$ and

$$\mathcal{R}(v) = \mathbb{E}((Y - u(X))^2) + \mathbb{E}((u(X) - v(X))^2),$$

or, by denoting μ the probability measure induced by X,

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

Therefore, the risk minimization over F is equivalent to the best approximation problem

$$\min_{v \in F} \|u - v\|^2_{L^2_\mu}$$

Bias and variance tradeoff

Denoting by u_F the minimizer of the risk $\mathcal{R}$ over F, the error (or excess risk)

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

For a given sample, when taking larger and larger model classes F

- approximation error $\searrow$
- but estimation error $\nearrow$

A suitable model class has to be selected in order to balance these two errors.

For high dimension d, or when only a small sample is available, one should use a model class of moderate dimension that exploits low-dimensional structures of function u.

This will require strategies for adaptation and selection of tensor formats.

Multilinear models

A tensor format (such as canonical format, tree-based format) is a multilinear model class, i.e. a set of functions having a multilinear parametrization

$$F = \{v(x) = \Psi(x)(a^1, \ldots, a^M) : a^\alpha \in \mathbb{R}^{K^\alpha}, 1 \leq \alpha \leq M\},$$

where a^α denotes a parameter in some vector space $\mathbb{R}^{K^\alpha}$, and

$$\Psi(x) : \mathbb{R}^{K^1} \times \ldots \times \mathbb{R}^{K^M} \to \mathbb{R}$$

is a multilinear map.

For a given α and fixed parameters $\{a^\beta : \beta \neq \alpha\}$, the partial map

$$a^\alpha \in \mathbb{R}^{K^\alpha} \mapsto \Psi(x)(a^1, \ldots, a^\alpha, \ldots, a^M) \in \mathbb{R}$$

is a linear map identified with a vector $\Psi^\alpha(x)$ in $\mathbb{R}^{K^\alpha}$ such that

$$\Psi(x)(a^1, \ldots, a^\alpha, \ldots, a^M) = \Psi^\alpha(x)^T a^\alpha = \sum_{k \in K^\alpha} \Psi^\alpha_k(x) a^\alpha_k.$$

Empirical risk minimization by alternating minimization

For a multilinear model class F, the empirical risk minimization

$$\min_{v \in F} \widehat{\mathcal{R}}_n(v) = \min_{a^1, \ldots, a^M} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \Psi(x_i)(a^1, \ldots, a^M))$$

can be solved with an alternating minimization algorithm (block coordinate descent), solving at each step

$$\min_{a^\alpha} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \Psi(x_i)(a^1, \ldots, a^\alpha, \ldots, a^M)) = \min_{a^\alpha} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \Psi^\alpha(x_i)^T a^\alpha) \qquad (\star)$$

for fixed parameters $\{a^\beta : \beta \neq \alpha\}$, which is a learning problem with a linear model.

For least-squares regression, $(\star)$ can be written

$$\min_{a^\alpha} \frac{1}{n} \|y - \mathbf{\Psi}^\alpha a^\alpha\|_2^2$$

where $y \in \mathbb{R}^n$ and $\mathbf{\Psi}^\alpha \in \mathbb{R}^n \otimes \mathbb{R}^{K^\alpha}$ is a matrix whose i-th row is $\Psi^\alpha(x_i)$.

Regularization

At each step, we can consider a regularized empirical risk minimization problem

$$\min_{a^\alpha} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \Psi^\alpha(x_i)^T a^\alpha) + \Omega_\alpha(a^\alpha) \qquad (\star)$$

with a regularization functional Ω_α promoting

- smoothness,
- sparsity (e.g. $\Omega_\alpha(a^\alpha) = \lambda_\alpha \|a^\alpha\|_1$ for convex relaxation methods, or a characteristic function for working set algorithms),
- ...

For least-squares regression and $\Omega_\alpha(a^\alpha) = \lambda_\alpha \|a^\alpha\|_1$, $(\star)$ is a LASSO problem

$$\min_{a^\alpha} \frac{1}{n} \|y - \mathbf{\Psi}^\alpha a^\alpha\|_2^2 + \lambda_\alpha \|a^\alpha\|_1$$

Cross-validation methods can be used for the selection of Ω_α.

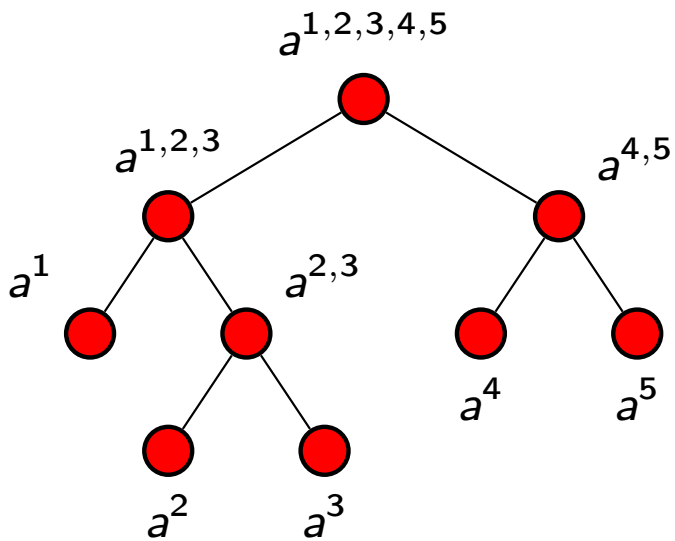
Parametrization of tree-based formats

We consider a tensor space $V = V^1 \otimes \ldots \otimes V^d$ of functions in $L^2_\mu(\mathcal{X})$, where $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ is equipped with a measure μ. We let $\{\phi^\nu_{i_\nu} : i_\nu \in I^\nu\}$ be a basis of the vector space V^ν in $L^2_{\mu_\nu}(\mathcal{X}_\nu)$, typically polynomials, wavelets, trigonometric polynomials...

We consider the model class of tree-based tensors $\mathcal{T}_r^T = \{v \in V : \text{rank}_T(v) \leq r\}$ where T is a dimension partition tree and r a tuple of ranks.

A function v in $\mathcal{T}_r^T$ admits the following multilinear parametrization in terms of low-order tensors a^α

$$v(x) = \sum_{\substack{i_\alpha \in I^\alpha \\ \alpha \in \mathcal{L}(T)}} \sum_{\substack{1 \leq k_\beta \leq r_\beta \\ \beta \in T}} \prod_{\alpha \in T \setminus \mathcal{L}(T)} a^\alpha_{(k_\beta)_{\beta \in S(\alpha)}, k_\alpha} \prod_{\alpha \in \mathcal{L}(T)} a^\alpha_{i_\alpha, k_\alpha} \phi^\alpha_{i_\alpha}(x_\alpha) = \Psi(x)((a^\alpha)_{\alpha \in T})$$



Parametrization of tree-based formats

A tensor v in tree-based format $\mathcal{T}_r^T$ admits infinitely many equivalent representations.

For a given α, it is possible to obtain a representation

$$v(x) = \Psi^\alpha(x)(a^\alpha) = \sum_k \Psi_k^\alpha(x) a_k^\alpha.$$

where $\{\Psi_k^\alpha(x)\}$ is a set of orthonormal functions in L^2_μ (depending on the parameters a^β, $\beta \neq \alpha$), i.e. such that

$$\mathbb{E}(\Psi_k^\alpha(X)\Psi_l^\alpha(X)) = \delta_{k,l}$$

This improves numerical stability and statistical properties and cross-validation error estimations.

For a least-square problem, empirical risk minimisation over a^α is equivalent to solving the linear system of equations

$$\frac{1}{n}\boldsymbol{\Psi}^{\alpha\,T}\boldsymbol{\Psi}^\alpha a^\alpha = \frac{1}{n}\boldsymbol{\Psi}^{\alpha\,T} y$$

where $\frac{1}{n}(\boldsymbol{\Psi}^{\alpha\,T}\boldsymbol{\Psi}^\alpha)_{k,l} = \frac{1}{n}\sum_{i=1}^n \Psi_k^\alpha(x_i)\Psi_l^\alpha(x_i)$ is an estimation of $\mathbb{E}(\Psi_k^\alpha(X)\Psi_l^\alpha(X)) = \delta_{k,l}$.

How to choose the ranks ?

For a given dimension tree T and a tuple of ranks $r = (r_\alpha)_{\alpha \in T}$,

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

Therefore

$$\mathcal{T}_r^T \subset \mathcal{T}_m^T \quad \text{for} \quad r \leq m,$$

and

$$\bigcup_{r \in \mathbb{N}^d} \mathcal{T}_r^T \quad \text{is dense in (or equal to)} \quad V$$

so that we can find a sequence of approximations u_r with increasing rank r which converges to any function in V.

We would like to find an increasing sequence of T-ranks which yields an optimal convergence of the error in terms of the complexity, and hopefully achieves optimal statistical performance in a supervised learning context.

Strategy for rank adaptation

- Construction of a sequence of approximations in tree-based format

$$u^m \in \mathcal{T}_{r^m}^T = \{v : rank_T(v) \leq (r_\alpha^m)_{\alpha \in T}\}$$

with increasing ranks

$$\begin{cases} r_\alpha^{m+1} = r_\alpha^m + 1 & \text{if } \alpha \in T_m^\theta \\ r_\alpha^{m+1} = r_\alpha^m & \text{if } \alpha \notin T_m^\theta \end{cases}$$

where T_m^θ is a suitably chosen subset of nodes.

- If we knew the truncation errors

$$\varepsilon_{r_\alpha^m}^\alpha(u) = \min_{\mathrm{rank}_\alpha(v) \leq r_\alpha^m} \mathcal{R}(v) - \mathcal{R}(u)$$

we could choose

$$T_m^\theta = \left\{ \alpha : \varepsilon_{r_\alpha^m}^\alpha(u) \geq \theta \max_{\beta \in T} \varepsilon_{r_\beta^m}^\beta(u) \right\}$$

for some $\theta \in [0, 1]$.

- With $\theta = 0$, all ranks are increased, while with $\theta = 1$, only the ranks with maximal truncation error are increased.

Strategy for rank adaptation

- For least-squares regression,

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

The α-matricisation of u admits a singular value decomposition

$$u(x) = \sum_{k \geq 1} \sigma_k^\alpha u_k^\alpha(x_\alpha) u_k^{\alpha^c}(x_{\alpha^c})$$

with singular values $\{\sigma_k^\alpha\}$ sorted by decreasing values, and the truncation error in L^2_μ-norm is

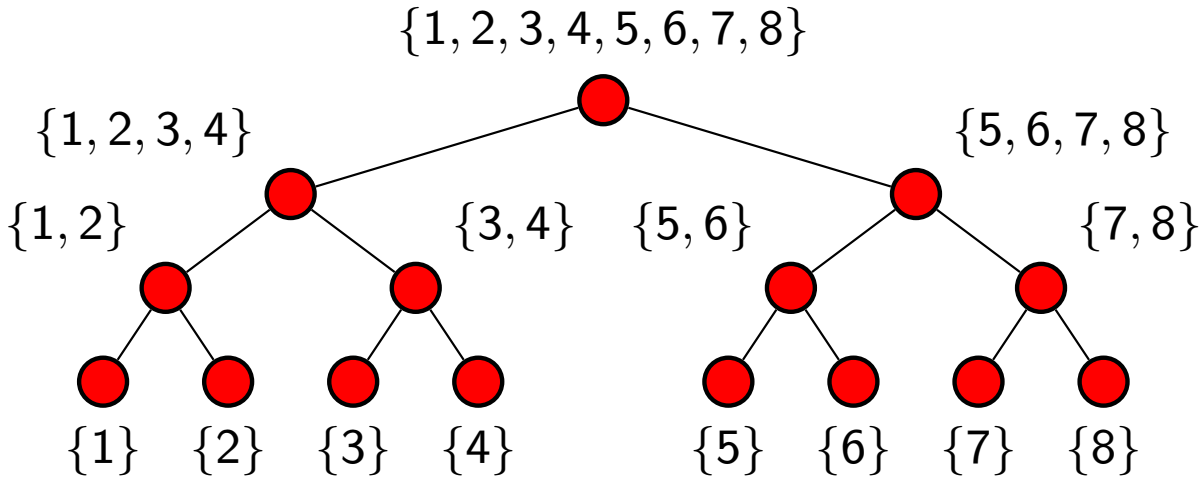
$$\varepsilon^\alpha_{r^m_\alpha}(u) = \min_{\mathrm{rank}_\alpha(v) \leq r^m_\alpha} \|u - v\|^2_{L^2_\mu} = \sum_{k > r^m_\alpha} (\sigma_k^\alpha)^2.$$

- In practice, at iteration m, we compute a low-rank correction of u^m to obtain a better approximation $\tilde{u}$ of u and estimate $\varepsilon^\alpha_{r^m_\alpha}(u)$ by $\varepsilon^\alpha_{r^m_\alpha}(\tilde{u})$.

Illustration

$$u(X_1, \ldots, X_8) = (1 + 5^{-1}X_1 + 5^{-2}X_3 + 5^{-3}X_5 + 5^{-4}X_7)^{-2}, \quad X_k \sim U(-1, 1) \ (i.i.d.)$$

- Least-squares regression
- Training sample of size $n = 100$
- Polynomial approximation spaces $V_\nu = \mathbb{P}_{10}$.
- Dimension tree



- Test sample of size 1000.

Illustration

- $\theta = 0$ (isotropic ranks)

iteration	rank	test error	CV estimate
0	(1 1 1 1 1 1 1 1 1 1 1 1 1 1)	$6.5\,10^{-3}$	$4.9\,10^{-3}$
1	(2 2 2 2 2 2 2 2 2 2 2 2 2 2)	$6.4\,10^{-5}$	$7.6\,10^{-6}$
3	(3 3 3 3 3 3 3 3 3 3 3 3 3 3)	$5.8\,10^{-5}$	$2.4\,10^{-8}$
4	(4 4 4 4 4 4 4 4 4 4 4 4 4 4)	$5.9\,10^{-5}$	$9.6\,10^{-16}$

- $\theta = 0.8$ (anisotropic ranks)

iteration	rank	test error	CV estimate
0	(1 1 1 1 1 1 1 1 1 1 1 1 1 1)	$6.5\,10^{-3}$	$4.9\,10^{-3}$
1	(1 1 2 2 1 1 2 1 2 1 1 1 1 1)	$1.9\,10^{-3}$	$7.7\,10^{-4}$
2	(2 2 2 2 2 2 2 1 2 1 2 1 2 1)	$2.8\,10^{-4}$	$1.9\,10^{-4}$
3	(2 2 2 2 2 2 2 1 2 1 2 1 2 1)	$5.9\,10^{-5}$	$8.2\,10^{-6}$
4	(2 2 3 3 2 2 3 1 3 1 2 1 2 1)	$3.3\,10^{-6}$	$3.8\,10^{-7}$
5	(3 3 3 3 3 2 3 1 3 1 3 1 2 1)	$1.0\,10^{-6}$	$4.9\,10^{-8}$
6	(3 3 4 3 3 2 4 1 3 1 3 1 2 1)	$7.2\,10^{-7}$	$3.0\,10^{-8}$
7	(3 3 5 4 3 2 5 1 4 1 3 1 2 1)	$1.2\,10^{-6}$	$3.8\,10^{-9}$

- In practice, automatic selection of $\theta > 0$.

Tree adaptation

Choosing a good dimension tree T is a combinatorial problem.

Exploring the whole set of possible trees in unfeasible in high dimension.

A possible strategy is to use a stochastic optimization algorithm over the set of possible trees, where at each iteration,

1. a tree T is randomly selected,
2. an approximation v is constructed in $\mathcal{T}_r^T$ (with an adapted rank),
3. a criterium (some error estimate) is used for accepting or rejecting the tree T.

Stochastic optimization may require many iterations, with a very costly step 2.

Tree adaptation

In practice, we use a slightly different approach using an efficient stochastic tree optimization algorithm for reducing the complexity of the representation of a given tensor.

We start with an initial tree T and learn an approximation $v \in \mathcal{T}_r^T$ with rank $r = (1, ..., 1)$. Then we repeat the following steps

- estimate truncation errors $\epsilon_{r_\alpha}^{\alpha}(u)$ and increase the rank r according a rank adaptation rule,
- learn an approximation v in the format $\mathcal{T}_r^T$ (with a good initialization),
- optimize the tree for reducing the storage complexity of v (using a stochastic algorithm): if a better tree T' is found, change the representation of v and set $r = \text{rank}_{T'}(v)$ and $T = T'$.

Illustration: sum of bivariate functions

$$u(X) = g(X_1, X_2) + g(X_3, X_4) + \ldots + g(X_{d-1}, X_d), \quad g(t, s) = \sum_{i=0}^{3} t^i s^i$$

We consider least-squares regression and use polynomial spaces $V_\nu = \mathbb{P}_3$ (no discretization error).

The function u admits an exact representation in the following tree T^1 with an optimal storage complexity 428.

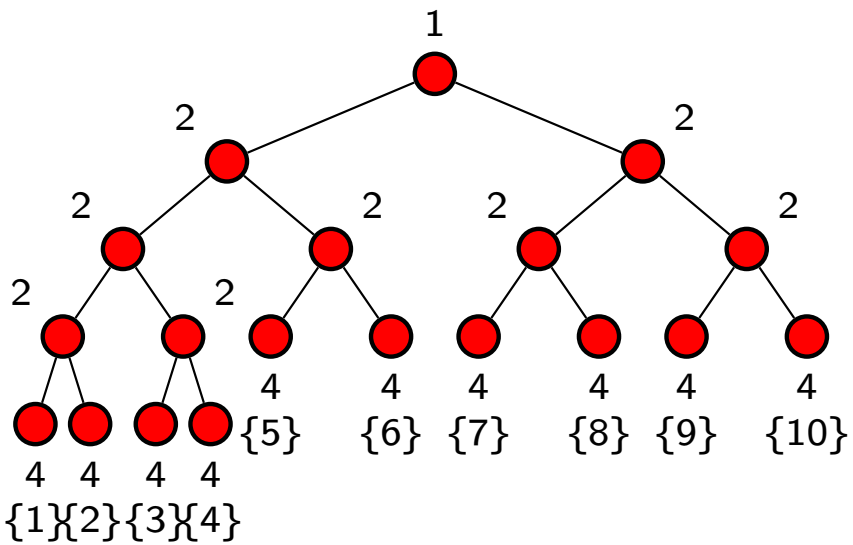


Figure: Tree T^1 and corresponding α-ranks.

Illustration: sum of bivariate functions

The function u admits an exact representation in the following tree T^2 with a higher storage complexity 560.

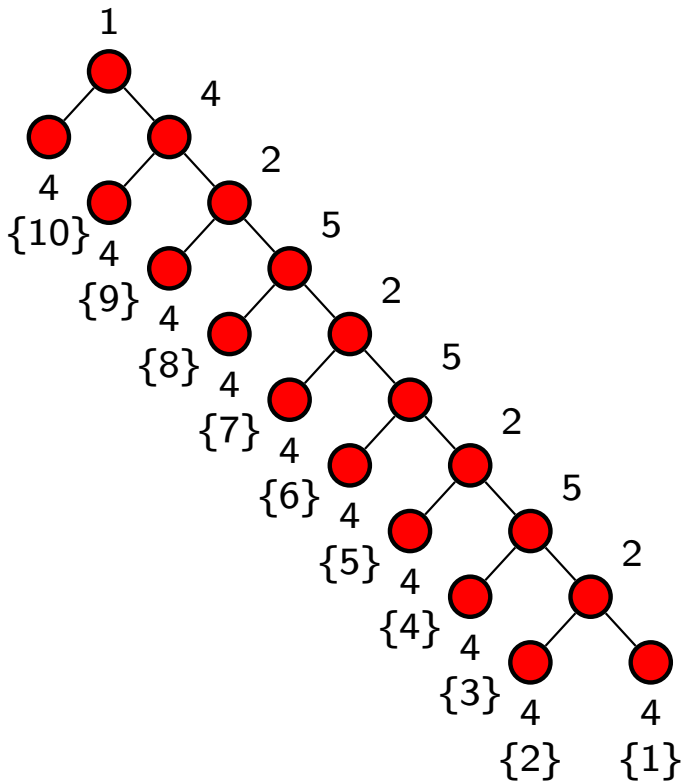


Figure: Tree T^2 and corresponding α-ranks.

Noiseless case, $Y = u(X)$

We start the learning algorithm with a random permutation $T^i_\sigma = \{\sigma(\alpha) : \alpha \in T^i\}$ of the tree T^i.

initial tree	n	$\hat{\mathbb{P}}(T$ is optimal)	$\varepsilon_{test}(v)$	$C(T, r)$
T^1_σ	$5\,10^2$	50%	$[4.23\,10^{-15}, 1.80\,10^{-1}]$	$[84, 921]$
	10^3	100%	$[6.64\,10^{-16}, 9.60\,10^{-15}]$	$[428, 673]$
	10^4	100%	$[5.34\,10^{-16}, 1.18\,10^{-15}]$	$[428, 428]$
T^2_σ	$5\,10^2$	70%	$[5.83\,10^{-15}, 1.94\,10^{-1}]$	$[69, 1114]$
	10^3	90%	$[7.72\,10^{-16}, 2.43\,10^{-2}]$	$[357, 515]$
	10^4	100%	$[5.59\,10^{-16}, 1.74\,10^{-15}]$	$[428, 428]$

Table: training sample size n, estimation of the probability of obtaining an optimal tree and ranges (over the 10 trials) for the test error, and the storage complexity.

Noisy case, $Y = u(X) + \varepsilon$, $Var(\varepsilon) = \zeta^2$

We here use the tree adaptation, starting from a random permutation T^2_σ of tree T^2.

n	ζ	$\mathbb{P}$(optimal T)	$\varepsilon_{test}(v)$	$C(T, v)$
10^3	10^{-1}	90%	$[9.4\,10^{-3}, 3.8\,10^{-2}]$	$[298, 450]$
	10^{-2}	80%	$[7.7\,10^{-4}, 1.2\,10^{-1}]$	$[114, 718]$
	10^{-3}	100%	$[7.5\,10^{-5}, 3.1\,10^{-2}]$	$[272, 570]$
5.10^3	10^{-1}	100%	$[5.5\,10^{-3}, 7.3\,10^{-3}]$	$[298, 545]$
	10^{-2}	100%	$[2.8\,10^{-4}, 3.5\,10^{-4}]$	$[428, 428]$
	10^{-3}	100%	$[2.9\,10^{-5}, 3.6\,10^{-5}]$	$[428, 428]$
10^4	10^{-1}	100%	$[1.8\,10^{-3}, 5.7\,10^{-3}]$	$[428, 570]$
	10^{-2}	100%	$[1.8\,10^{-4}, 2.3\,10^{-4}]$	$[428, 428]$
	10^{-3}	100%	$[1.9\,10^{-5}, 2.4\,10^{-5}]$	$[428, 428]$

Table: Training sample size n, standard deviation ζ of the noise, probability of obtaining an optimal tree T and confidence intervals for the test error, and the storage complexity.

Noisy case, $Y = u(X) + \varepsilon$, $Var(\varepsilon) = \zeta^2$

$$\mathcal{R}(v) = \mathcal{R}(u) + \|u - v\|^2 = \zeta^2 + \|u - v\|^2$$

n	ζ^2	$\|u - v\|^2$
10^3	10^{-2}	$[2.90\,10^{-3}, 4.68\,10^{-2}]$
	10^{-4}	$[1.90\,10^{-5}, 0.47\,10^{-1}]$
	10^{-6}	$[1.81\,10^{-7}, 0.03\,10^{-2}]$
$5.\,10^3$	10^{-2}	$[1.00\,10^{-3}, 1.72\,10^{-3}]$
	10^{-4}	$[2.52\,10^{-6}, 4.08\,10^{-6}]$
	10^{-6}	$[2.73\,10^{-8}, 4.32\,10^{-8}]$
10^4	10^{-2}	$[1.11\,10^{-4}, 1.04\,10^{-3}]$
	10^{-4}	$[1.13\,10^{-6}, 1.75\,10^{-6}]$
	10^{-6}	$[1.18\,10^{-8}, 1.85\,10^{-8}]$

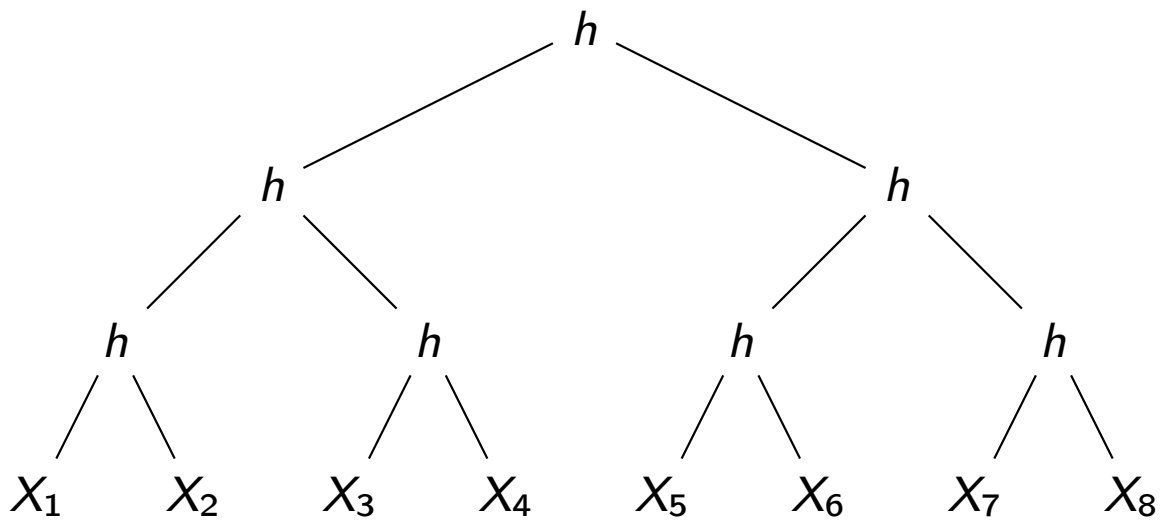
Table: Training sample size n, standard deviation ζ of the noise and confidence interval for the estimated approximation error.

Illustration: composition of functions

Consider a tree-structured composition of functions

$$u(X) = h(h(h(X_1, X_2), h(X_3, X_4)), h(h(X_5, X_6), h(X_7, X_8))),$$

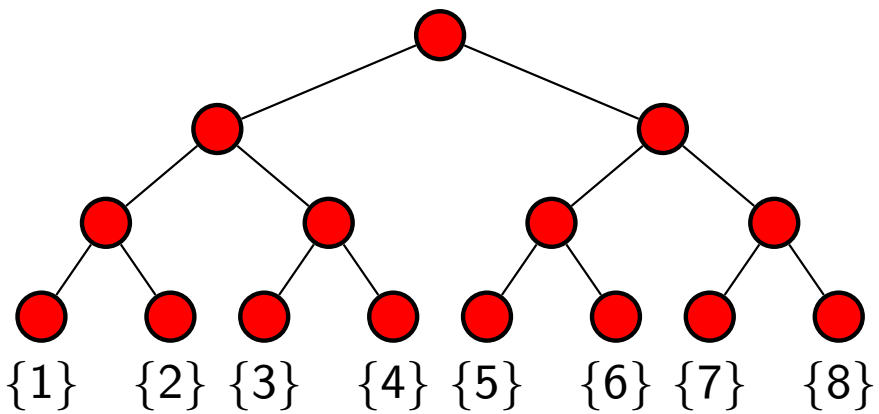
where $h(t, s) = 9^{-1}(2 + ts)^2$ is a bivariate function and where the $d = 8$ random variables $X_1, \ldots, X_8$ are independent and uniform on $[-1, 1]$.



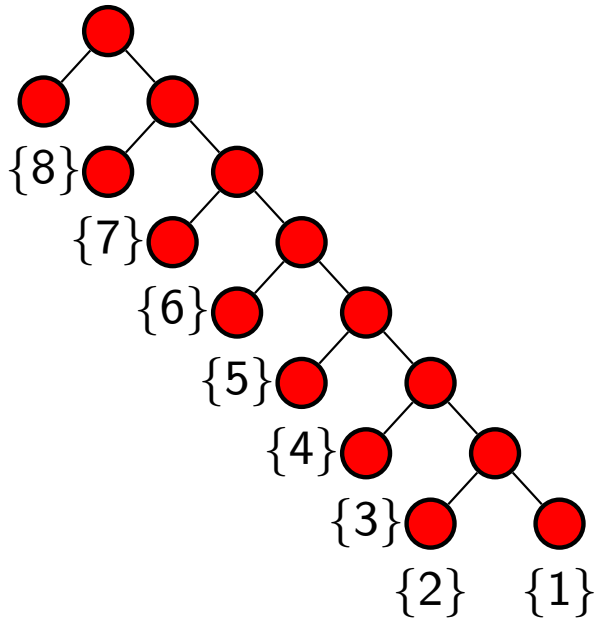
We use approximation spaces $V^\nu = \mathbb{P}_8(\mathcal{X}_\nu)$, so that function u is in V and could (in principle) be recovered exactly for any choice of tree with a sufficiently high rank.

Illustration: composition of functions

We consider two trees T^1 (coinciding with the structure of u) and T^2.



(a) Tree T^1



(b) Tree T^2

Illustration: composition of functions

We start the learning algorithm from the tree and the associated families of trees $T^2_\sigma = \{\sigma(\alpha) : \alpha \in T^2\}$ obtained by applying a random permutation σ to T^2.

n	$\hat{\mathbb{P}}(T = T^1)$	$\varepsilon_{test}(v)$	$C(T, r)$
10^3	90%	$[1.75\,10^{-5}, 1.75\,10^{-4}]$	$[360, 1062]$
10^4	90%	$[2.15\,10^{-8}, 4.10\,10^{-3}]$	$[185, 2741]$
10^5	100%	$[4.67\,10^{-15}, 8.92\,10^{-3}]$	$[163, 2594]$

Table: training sample size n, estimation of the probability of obtaining the ideal tree T^1 and ranges (over the 10 trials) for the test error, and the storage complexity.

Behavior of the algorithm algorithm with $n = 10^5$

Iteration	rank r	$\varepsilon_{test}(v)$	$C(T, r)$
1	$(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)$	$3.38\,10^{-2}$	79
2	$(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1)$	$2.95\,10^{-2}$	100
3	$(1,1,2,1,1,1,1,1,1,1,1,1,2,1,1)$	$2.95\,10^{-2}$	100
4	$(1,1,2,1,2,1,1,1,2,1,1,1,2,1,1)$	$2.45\,10^{-2}$	121
5	$(1,1,2,1,2,1,1,1,2,1,1,1,2,1,1)$	$2.45\,10^{-2}$	121
6	$(1,1,2,1,2,1,1,1,2,1,2,1,2,2,1)$	$1.85\,10^{-2}$	142
7	$(1,1,2,1,2,1,1,1,2,1,2,1,2,2,1)$	$1.85\,10^{-2}$	142
8	$(1,1,2,1,2,1,2,1,2,1,2,1,2,2,2)$	$8.97\,10^{-3}$	163
9	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$9.54\,10^{-3}$	188
10	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$8.89\,10^{-3}$	188
11	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$9.47\,10^{-3}$	188
12	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$8.87\,10^{-3}$	188
13	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$5.22\,10^{-3}$	188
14	$(1,2,2,2,2,2,2,2,2,1,2,2,2,2,2)$	$3.97\,10^{-3}$	188
15	$(1,2,3,2,3,2,3,2,3,2,3,2,3,3,3)$	$1.55\,10^{-4}$	308
16	$(1,3,3,3,3,2,3,3,3,2,3,3,3,3,3)$	$1.18\,10^{-4}$	364
17	$(1,3,3,3,3,2,3,3,3,2,3,3,3,3,3)$	$1.18\,10^{-4}$	364

Outline

Setting

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.

Singular value decomposition

Consider a subset of variables α and its complementary subset $\alpha^c = D \setminus \alpha$.

A multivariate function $u(x_1, \ldots, x_d)$ is identified with a bivariate function $u \in V_\alpha \otimes V_{\alpha^c}$ which admits a singular value decomposition

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

The problem of best approximation of u by a function with α-rank r_α,

$$\min_{\mathrm{rank}_\alpha(v) \leq r_\alpha} \|u - v\|^2,$$

admits as a solution the truncated singular value decomposition u_{r_α} of u

$$u_{r_\alpha}(x_\alpha, x_{\alpha^c}) = \sum_{k=1}^{r_\alpha} \sigma_k^\alpha v_k^\alpha(x_\alpha) v_k^{\alpha^c}(x_{\alpha^c})$$

where $\{v_1^\alpha, \ldots, v_{r_\alpha}^\alpha\}$ are the r_α α-principal components of u.

α-principal subspace and associated projection

The subspace of principal components

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

is such that

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

where $\mathcal{P}_{U_\alpha}$ is the orthogonal projection onto U_α.

It is solution of

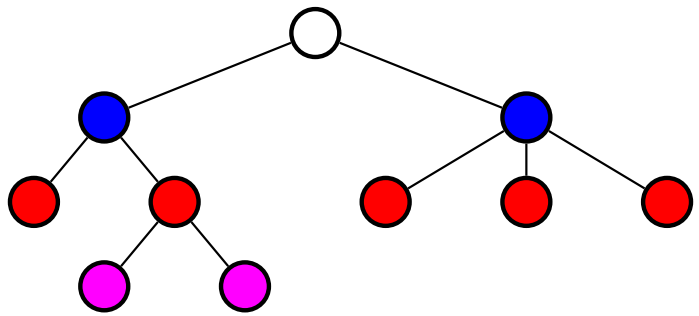
$$\min_{\dim(U_\alpha)=r_\alpha} \|u - \mathcal{P}_{U_\alpha} u\|^2$$

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

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

Higher-order principal component analysis for tree-based formats

Let T be a tree-structured collection of subsets of 2^D



For each α in T, we will determine subspaces U_α that are approximations of α-principal subspaces of u in low-dimensional subspaces V_α of functions defined on $\mathcal{X}_\alpha$.

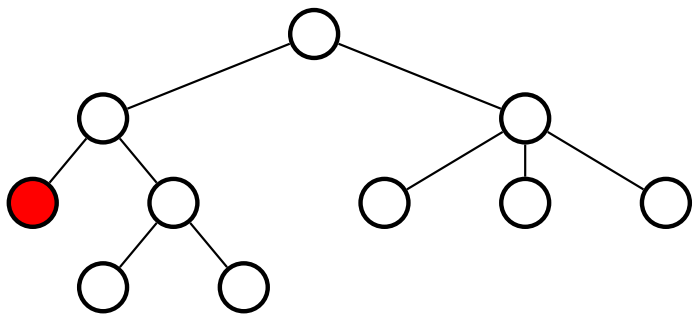
Higher-order principal component analysis for tree-based formats

For each $\alpha \in T \setminus D$, U_α is defined as the r_α-dimensional α-principal subspace of

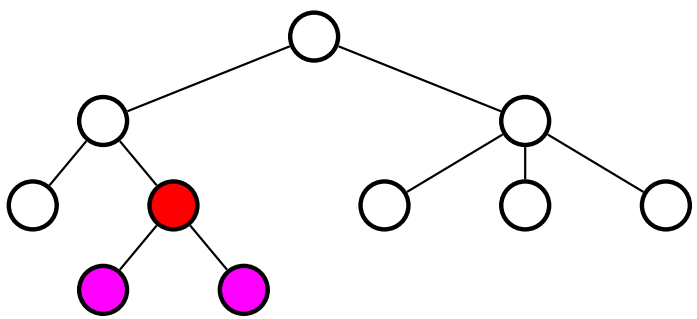
$$u_\alpha(\cdot, x_{\alpha^c}) = \mathcal{P}_{V_\alpha} u(\cdot, x_{\alpha^c})$$

- for $S(\alpha) = \emptyset$ (leaf node), V_α is a given approximation space (e.g., polynomials, wavelets, kernel functions, perceptrons...)

$$V_\alpha = \text{span}\{\phi_\lambda^\alpha(X_\alpha) : \lambda \in I^\alpha\}$$



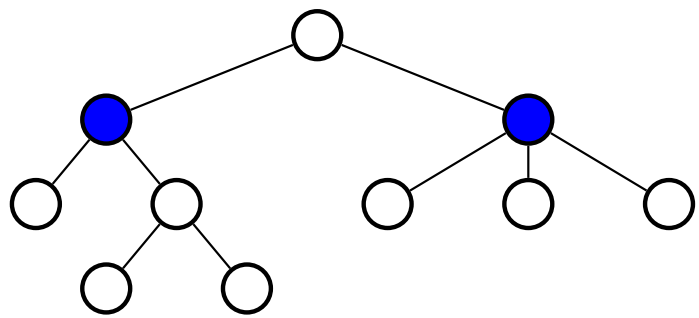
- for $S(\alpha) \neq \emptyset$ (interior node), $V_\alpha = \bigotimes_{\beta \in S(\alpha)} U_\beta$.



Higher-order principal component analysis for tree-based formats

We finally obtain an approximation $u^\star$ of u by orthogonal projection onto the tensor space $V_D = \bigotimes_{\alpha \in S(D)} U_\alpha$

$$u^\star = \mathcal{P}_{V_D} u$$



Theorem (Fixed rank)

For a given T-rank, we obtain an approximation $u^\star \in \mathcal{T}_r^T$ such that

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

Theorem (Fixed precision)

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

$$\|\mathcal{P}_{U_\alpha} u_\alpha - u_\alpha\| \leq \frac{\epsilon}{\sqrt{\#T}} \|u_\alpha\|,$$

we obtain an approximation $u^\star$ such that

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

Learning algorithm based on principal component analysis

For a feasible algorithm using samples...

1. From orthogonal to sampled-based projections.
2. 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, use of magic points x_α^i,
- for least-squares projection, possible use of optimal weighted least-squares.

Statistical estimation of principal subspaces

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

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

where u is seen as a function-valued random variable

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

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

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

Properties of the algorithm (for interpolation)

Theorem (Prescribed rank)

For a given T-rank, if the subspaces U_α are such that

$$\|\mathcal{P}_{U_\alpha} u_\alpha - u_\alpha\| \leq C \min_{\mathrm{rank}_\alpha(v) \leq r_\alpha} \|v - u_\alpha\|$$

then we obtain an approximation $u^\star$ such that

$$\|u^\star - u\|^2 \leq \Lambda^2 C^2 \# T \min_{v \in \mathcal{T}_r^T} \|v - u\|^2 + \tilde{\Lambda}^2 \max_{1 \leq \nu \leq d} \|u - \mathcal{P}_{V_\nu} u\|^2$$

with Λ and $\tilde{\Lambda}$ depending on the properties of the oblique projection operators.

About complexity: If $N_\alpha = r_\alpha$ for all $\alpha \in T$, then the total number of evaluations N is equal to the storage complexity S of the resulting approximation $u^\star \in \mathcal{T}_r^T$.

About the constants

The constants Λ and $\tilde{\Lambda}$ depend on

$$\|I_{V_\alpha}\|_{U_\alpha^{min}(u)\to V_\alpha} \quad \text{and} \quad \|I_{U_\alpha} - P_{U_\alpha}\|_{U_\alpha^{min}(u)\to V_\alpha}$$

which depends on the properties of interpolation operators restricted to minimal subspaces of u.

Case of tensor recovery

Assume that $U_\alpha^{min}(u) \subset V_\alpha$ for all leaves α (no discretization error).

If for all $\alpha \in T$, the set of N_α samples $u(\cdot, x_{\alpha^c}^k)$ contains $\text{rank}_\alpha(u)$ linearly independent functions, then $U_\alpha = U_\alpha^{min}(u)$.

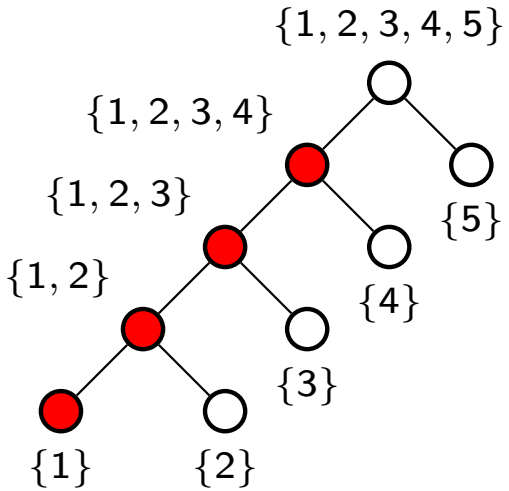
The constants $C = 1$, $\Lambda = 1$, and $\tilde{\Lambda} = 1$ (i.e. same stability than the ideal algorithm).

Illustration of tensor recovery: Henon-Heiles potential

$$u(X) = \frac{1}{2}\sum_{i=1}^{d} X_i^2 + 0.2\sum_{i=1}^{d-1}(X_i X_{i+1}^2 - X_i^3) + \frac{0.2^2}{16}\sum_{i=1}^{d-1}(X_i^2 + X_{i+1}^2)^2, \quad X_i \sim U(-1,1),$$

$\text{rank}_\alpha(u) = 3$ for all α in

$T = \{\{1\}, \{1,2\}, \ldots, \{1,\ldots,d-1\}\}$



Then u can be exactly represented in the tensor train format $\mathcal{T}_r^T$ with T-rank $r = (3, \ldots, 3)$

$$u = \sum_{k_1=1}^{3}\sum_{k_2=1}^{3}\cdots\sum_{k_{d-1}=1}^{3} v_{1,k_1}^{(1)}(x_1)\, v_{k_1,k_2}^{(1,2)}(x_2)\, v_{k_2,k_3}^{(1,2,3)}(x_3)\ldots v_{k_{d-1},1}^{(1,\ldots,d)}(x_d)$$

with univariate polynomial functions of degree 4.

Illustration of tensor recovery: Henon-Heiles potential

Table: Approximation with prescribed T-rank $r = (3, \ldots, 3)$ and polynomial degree 4 for different values of d and $\gamma = N_\alpha / r_\alpha$. Use of interpolation.

$\gamma = 1$					
d	5	10	20	50	100
$\varepsilon(u^\star) \times 10^{14}$	[1.0; 234.2]	[1.5; 67.5]	[2.5; 79.9]	[6.6; 62.8]	[15.7; 175.1]
$S = N$	165	390	840	2190	4440

$\gamma = 10$					
d	5	10	20	50	100
$\varepsilon(u^\star) \times 10^{14}$	[0.1; 0.4]	[0.2; 0.4]	[0.3; 0.4]	[0.4; 0.7]	[0.6; 0.8]
S	165	390	840	2190	4440
N	1515	3765	8265	21765	44265

Properties of the algorithm (for interpolation)

Theorem (Fixed precision)

Let $\epsilon, \tilde{\epsilon} \geq 0$. If the subspaces U_α are determined such that

$$\|\mathcal{P}_{U_\alpha} u_\alpha - u_\alpha\| \leq \frac{\epsilon}{\sqrt{\#T}} \|u_\alpha\|$$

and if the approximation spaces V_ν, $1 \leq \nu \leq d$, are such that

$$\|\mathcal{P}_{V_\nu} u - u\| \leq \tilde{\epsilon} \|u\|,$$

then we obtain an approximation $u^\star$ such that

$$\|u^\star - u\|^2 \leq (\Lambda^2 \epsilon^2 + \tilde{\Lambda}^2 \tilde{\epsilon}^2) \|u\|^2$$

with Λ and $\tilde{\Lambda}$ depending on the properties of the oblique projection operators.

Illustration for approximation: Borehole function

The Borehole function models water flow through a borehole:

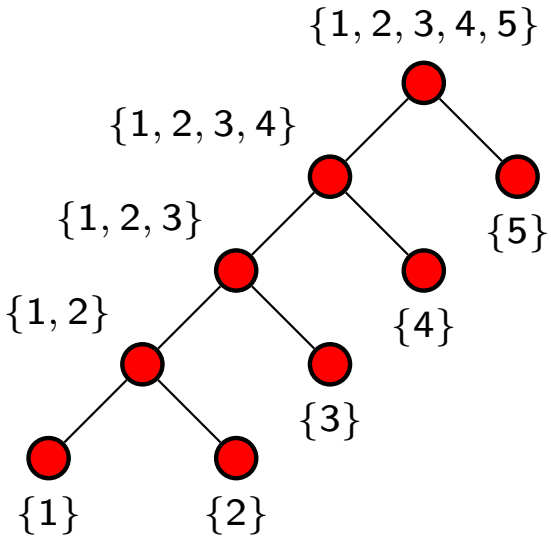
$$u(X) = \frac{2\pi T_u (H_u - H_l)}{\ln(r/r_w)\left(1 + \frac{2LT_u}{\ln(r/r_w) r_w^2 K_w} + \frac{T_u}{T_l}\right)}, \quad X = (r_w, \log(r), T_u, H_u, T_l, H_l, L, K_w)$$

r_w	radius of borehole (m)	$N(\mu = 0.10, \sigma = 0.0161812)$
r	radius of influence (m)	$LN(\mu = 7.71, \sigma = 1.0056)$
T_u	transmissivity of upper aquifer (m^2/yr)	$U(63070, 115600)$
H_u	potentiometric head of upper aquifer (m)	$U(990, 1110)$
T_l	transmissivity of lower aquifer (m^2/yr)	$U(63.1, 116)$
H_l	potentiometric head of lower aquifer (m)	$U(700, 820)$
L	length of borehole (m)	$U(1120, 1680)$
K_w	hydraulic conductivity of borehole (m/yr)	$U(9855, 12045)$

Illustration for approximation: Borehole function

Approximation in hierarchical Tucker format with a linearly structured tree:

$$T = \{\{1\}, \ldots, \{d\}, \{1,2\}, \ldots, \{1, \ldots, d-1\}, D\}$$



$$u^\star = \sum_{i_1=1}^{r_1} \ldots \sum_{i_d=1}^{r_d} \sum_{k_2=1}^{r_{1,2}} \ldots \sum_{k_{d-1}=1}^{r_{1,\ldots,d-1}} v_{i_1}^{(1)}(x_1) \ldots v_{i_d}^{(d)}(x_d) C_{i_1,i_2,k_2}^{(1,2)} C_{k_2,i_3,k_3}^{(1,2,3)} \ldots C_{k_{d-2},i_{d-1},k_{d-1}}^{(1,\ldots,d-1)} C_{k_{d-1},i_d}^{(1,\ldots,d}$$

with polynomial functions $v_{i_\nu}^{(\nu)} \in V_\nu = \mathbb{P}_q$.

Illustration for approximation: Borehole function

Table: Approximation with prescribed precision ϵ, adaptive degree $p(\epsilon) = \log_{10}(\epsilon^{-1})$, and $N_\alpha = \dim(V_\alpha)$. Confidence intervals for relative error $\varepsilon(u^\star)$, storage complexity S and number of evaluations M for different ϵ, and average ranks. Projections based on empirical interpolation

ϵ	$\varepsilon(u^\star)$	N	S	$[r_{\{1\}}, \ldots, r_{\{d\}}, r_{\{1,2\}}, \ldots, r_{\{1,\ldots,d-1\}}]$
10^{-1}	$[1.8; 2.7] \times 10^{-1}$	$[39, 39]$	$[23, 23]$	$[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$
10^{-2}	$[0.3; 4.0] \times 10^{-2}$	$[88, 100]$	$[41, 46]$	$[1,1,1,1,1,1,1,1,1,1,2,1,2,1,1]$
10^{-3}	$[0.8; 1.9] \times 10^{-3}$	$[159, 186]$	$[61, 78]$	$[2,1,1,2,2,1,1,1,1,2,2,2,1,1]$
10^{-4}	$[2.5; 5.6] \times 10^{-5}$	$[328, 328]$	$[141, 141]$	$[2,2,2,3,3,2,2,2,1,2,2,2,2,2]$
10^{-5}	$[0.6; 1.6] \times 10^{-5}$	$[444, 472]$	$[166, 178]$	$[2,2,2,4,4,2,2,2,1,2,2,2,2,2]$
10^{-6}	$[3.1; 5.7] \times 10^{-6}$	$[596, 664]$	$[204, 241]$	$[3,2,2,4,5,3,2,2,2,2,2,2,2,2]$
10^{-7}	$[1.0; 6.3] \times 10^{-7}$	$[1042, 1267]$	$[374, 429]$	$[4,3,4,6,5,3,3,3,2,2,3,2,2,2]$
10^{-8}	$[1.1; 7.1] \times 10^{-8}$	$[1567, 1567]$	$[512, 512]$	$[4,3,4,7,6,3,3,3,2,2,3,2,3,3]$
10^{-9}	$[0.2; 4.9] \times 10^{-8}$	$[1719, 1854]$	$[534, 560]$	$[4,4,4,8,6,3,3,3,2,2,3,2,3,3]$
10^{-10}	$[0.3; 1.9] \times 10^{-9}$	$[2482, 2828]$	$[774, 838]$	$[5,4,6,10,7,4,3,3,2,2,3,2,3,3]$

Influence of the tree

Table: Approximation with prescribed precision $\epsilon = 10^{-8}$, degree $p(\epsilon) = \log_{10}(\epsilon^{-1})$, and $N_\alpha = \dim(V_\alpha)$. Confidence intervals for relative error $\varepsilon(u^\star)$, storage complexity S and number of evaluations M for different ϵ, and average ranks. Projections based on empirical interpolation

ϵ	$\varepsilon(u^\star)$	N	S
10^{-8}	$[1.2; 1.7] \times 10^{-7}$	$[1587, 1587]$	$[527, 527]$

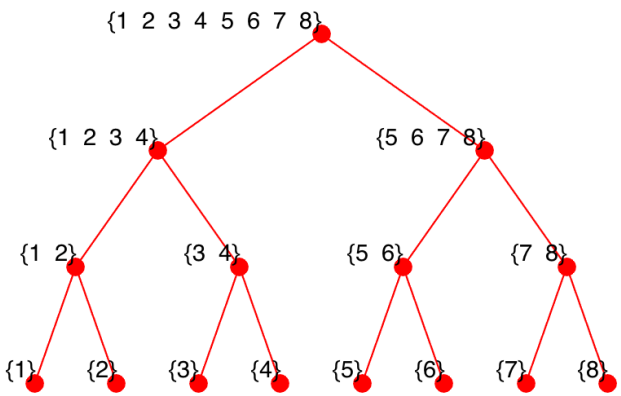


Illustration: Henon-Heiles potential ($d = 20$)

Table: Approximation with prescribed precision $\epsilon = 10^{-8}$, degree $p = 4$, and $N_\alpha = \dim(V_\alpha)$. Confidence intervals for relative error $\varepsilon(u^\star)$, storage complexity S and number of evaluations M, and average maximal rank.

ϵ	$\varepsilon(u^\star)$	N	S	$\max_\alpha r_\alpha$
10^{-8}	$[1.6e-14; 2.9e-14]$	$[3101, 3101]$	$[1047, 1047]$	4

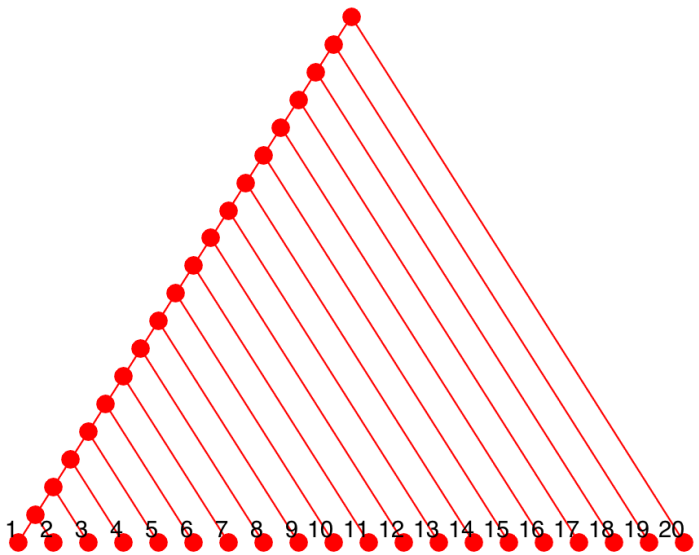


Table: Approximation with prescribed precision $\epsilon = 10^{-8}$, degree $p = 4$, and

Approximation of a function using tensorization

Consider a function $f : [0, 1] \to 1$ and the vector $v \in \mathbb{R}^{2^d}$ such that

$$v(i) = f(2^{-d} i), \quad 0 \leq i \leq 2^d - 1$$

The vector v can be identified with an order-d tensor $u \in \mathcal{H} = \mathbb{R}^2 \otimes \ldots \otimes \mathbb{R}^2$ such that

$$u(i_1, \ldots, i_d) = v(i), \quad i = \sum_{k=1}^{d} i_k 2^{d-k},$$

where $(i_1, \ldots, i_d) \in \{0, 1\}^d = \mathcal{X}$ is the binary representation of the integer i.

We introduce an approximation of u in the tensor train format

$$u(i_1, \ldots, i_d) \approx \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_{1,2}} \ldots \sum_{k_{d-1}=1}^{r_{1,\ldots,d-1}} v_{k_1}^{(1)}(i_1) v_{k_1,k_2}^{(2)}(i_2) \ldots v_{k_{d-2},k_{d-1}}^{(d-1)}(i_{d-1}) v_{k_{d-1}}^{(d)}(i_d)$$

Approximation of a function using tensorization

Table: $f(t) = \sqrt{t}$, $d = 40$. Approximation in tensor train format with prescribed ϵ, $N_\alpha = \dim(V_\alpha)$. Confidence intervals for relative ℓ^2-error $\varepsilon(u^\star)$, number of evaluations M, storage complexity S and maximal rank for different ϵ.

ϵ	$\varepsilon(u^\star)$	M	S	$\max_\alpha r_\alpha$
10^{-1}	$[9.3\,10^{-3}; 5.5\,10^{-2}]$	$[182, 230]$	$[90, 114]$	$[2, 2]$
10^{-2}	$[3.7\,10^{-3}; 8.6\,10^{-3}]$	$[314, 350]$	$[156, 172]$	$[2, 3]$
10^{-3}	$[5.4\,10^{-4}; 9.2\,10^{-4}]$	$[514, 606]$	$[252, 300]$	$[3, 3]$
10^{-4}	$[1.3\,10^{-4}; 3.3\,10^{-3}]$	$[838, 962]$	$[414, 474]$	$[4, 4]$
10^{-5}	$[1.8\,10^{-5}; 8.2\,10^{-4}]$	$[1270, 1398]$	$[626, 692]$	$[4, 5]$
10^{-6}	$[1.3\,10^{-6}; 6.3\,10^{-5}]$	$[1900, 2036]$	$[938, 1014]$	$[5, 5]$
10^{-7}	$[4.9\,10^{-7}; 1.3\,10^{-6}]$	$[2444, 2718]$	$[1218, 1344]$	$[5, 6]$
10^{-8}	$[1.0\,10^{-7}; 1.2\,10^{-6}]$	$[3304, 3468]$	$[1642, 1722]$	$[6, 6]$
10^{-9}	$[2.2\,10^{-8}; 1.3\,10^{-7}]$	$[4116, 4328]$	$[2046, 2144]$	$[7, 7]$
10^{-10}	$[8.6\,10^{-10}; 6.7\,10^{-8}]$	$[5024, 5136]$	$[2490, 2552]$	$[7, 7]$

References I

M. Chevreuil, R. Lebrun, A. Nouy, and P. Rai.

A least-squares method for sparse low rank approximation of multivariate functions.
SIAM/ASA Journal on Uncertainty Quantification, 3(1):897–921, 2015.

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.