# Statistical learning in tree-based tensor format

**Erwan Grelier** (GeM, Centrale Nantes) in collaboration with
**Mathilde Chevreuil** (GeM, Université de Nantes) and **Anthony Nouy** (LMJL, Centrale Nantes)

MASCOT-NUM Annual Conference | March 21, 2018

# Statistical learning for uncertainty quantification in high dimension

## Uncertainty quantification

Let $X = (X_1, \ldots, X_d)$ be a set of $d$ random variables modeling the (computational of experimental) uncertainties in a model $f$, and let $Y$ be a quantify of interest:

$$Y = f(X)\ .$$

Two types of problem: forward and inverse.

## Model order reduction for the uncertainty quantification

Uncertainty quantification requires the evaluation of the model $f$ for many instances of $X$. When evaluating the model is costly (computationally or experimentally), we rely on approximations $\tilde{f}$ of $f$ that are cheap to evaluate.

### Statistical learning

To compute $\tilde{f}$, minimize the empirical risk on a training sample $\left\{(x^k, y^k)\right\}_{k=1}^N$, where $y^k = f(x^k)$:

$$\min_{\tilde{f} \in \mathcal{V}} \frac{1}{N} \sum_{k=1}^N \ell(y^k, \tilde{f}(x^k))$$

where $\mathcal{V}$ is an approximation set, and with $\ell$ a loss function, e.g. the square loss function:

$$\ell(f(x^k), \tilde{f}(x^k)) = \left| f(x^k) - \tilde{f}(x^k) \right|^2 .$$

### High-dimensional approximation

When the dimension $d$ of $X$ is high, the approximation $\tilde{f}(X)$ is sought in sets of functions that are described by a number of parameters growing moderately with $d$.

# Outline

Tree-based tensor (TBT) formats

Statistical learning in tree-based tensor format

Tree-based tensor learning combined with changes of variables

# Tree-based tensor (TBT) formats

# Notions of rank

A multivariate function $v(x_1, \dots, x_d)$ can be identified with an order-$d$ tensor.

- A function with rank one:

$$v(x) = v_1(x_1) \dots v_d(x_d),$$

- a function with (canonical) rank r:

$$v(x) = \sum_{i=1}^{r} v_1^i(x_1) \dots v_d^i(x_d),$$

- a function with $\alpha$-rank $\text{rank}_\alpha(v) = r_\alpha$:

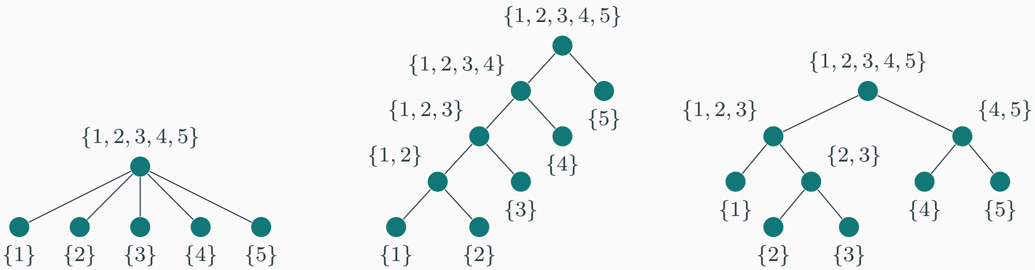
$$v(x) = \sum_{i=1}^{r_\alpha} v_\alpha^i(x_\alpha) v_{\alpha^c}^i(x_{\alpha^c}),$$

  with $x_\alpha$ and $x_{\alpha^c}$ complementary groups of variables.

- For $T \subset 2^{\{1,\dots,d\}}$, a function with $T$-rank $r = (r_\alpha)_{\alpha \in T}$:

$$v(x) = \sum_{i=1}^{r_\alpha} v^i_\alpha(x_\alpha) v^i_{\alpha^c}(x_{\alpha^c}), \quad \forall \alpha \in T$$

When $T$ is a dimension partition tree, $v$ has a tree-based tensor (TBT) format.




Tucker — Tensor Train Tucker — Hierarchical Tucker

# Tree-based tensor formats

A function in tree-based tensor format admits a parametrization with functions associated to each node $\alpha$ that are multilinear in groups of variables.

Example: hierarchical tucker tensor with $d = 5$:

$$v(x) = f_{1,2,3,4,5}(f_{1,2,3}(f_1(x_1), f_{2,3}(f_2(x_2), f_3(x_3))), f_{4,5}(f_4(x_4), f_5(x_5)))$$

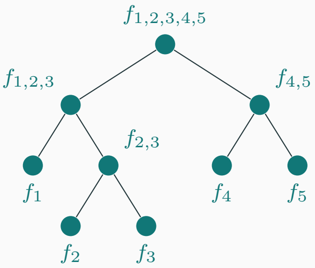
where for the leaves, $1 \leq \nu \leq d$,

$$f_\nu : \mathcal{X}^\nu \to \mathbb{R}^{r_\nu},$$

and e.g. for a node $\alpha$ with children $\beta_1$ and $\beta_2$,

$$f_\alpha : \mathbb{R}^{r_{\beta_1}} \times \mathbb{R}^{r_{\beta_2}} \to \mathbb{R}^{r_\alpha}$$

is a bilinear function identified with a tensor in $\mathbb{R}^{r_\alpha \times r_{\beta_1} \times r_{\beta_2}}$.




Hierarchical Tucker

Particular case of deep networks: see [Cohen, Sharir, and Shashua, 2015] [Khrulkov, Novikov, and Oseledets, 2017].

Properties of $\mathcal{T}_r^T = \{u : \text{rank}_\alpha(u) \leq r_\alpha, \alpha \in T\}$

- The storage complexity of $v \in \mathcal{T}_r^T$ scales as $O\left(dR^{s+1}\right)$, with $R = \max_{\alpha \in T} r_\alpha$ and $s$ the arity of the tree,
- $\mathcal{T}_r^T$ is a closed set: best approximation problems are well posed and stable algorithms exist,
- an element $v$ of $\mathcal{T}_r^T$ is linear in each of its parameters $f_\alpha$, enabling the use of the classical machinery of linear approximation in an alternating minimization algorithm,
- a higher-order singular value decomposition (HOSVD) of $v \in \mathcal{T}_r^T$ can be computed [Lathauwer, Moor, and Vandewalle, 2000].

# Statistical learning in tree-based tensor format

## Statistical learning in tree-based tensor format

The minimization problem

$$\min_{v \in \mathcal{T}_r^T} \frac{1}{N} \sum_{k=1}^{N} \ell\left(y^k, v(x^k)\right)$$

is solved using an alternating minimization algorithm.

Thanks to the linearity of $v$, each problem to solve is linear:

$$\min_{a_\alpha \in \mathbb{R}^{m_\alpha}} \frac{1}{N} \sum_{k=1}^{N} \ell\left(y^k, \Psi_\alpha(x^k)^{\mathsf{T}} a_\alpha\right),$$

with $\Psi_\alpha(x)$ such that $v(x) = \Psi_\alpha(x)^{\mathsf{T}} a_\alpha$, and with $m_\alpha = \begin{cases} r_\nu n_\alpha & \text{if } \alpha \text{ is a leaf,} \\ r_\alpha r_{\beta_1} \cdots r_{\beta_{s_\alpha}} & \text{otherwise.} \end{cases}$

For a leaf node $\nu \in T$, we compute a sparse approximation by using a working-set strategy. A cross validation estimator of the error is used to choose the optimal set.

## Adaptation of the tree-based ranks

- Start with a tree-based rank $r^0 = (0, \dots, 0)$,
- at iteration $m$, given $v_m \in \mathcal{T}^T_{r^m}$, select a subset of nodes to enrich $T_m \subseteq T$, and define $r^{m+1} = \left(r^{m+1}_\alpha\right)_{\alpha \in T}$ by

$$r^{m+1}_\alpha = \begin{cases} r^m_\alpha + 1 & \text{if } \alpha \in T_m, \\ r^m_\alpha & \text{if } \alpha \notin T_m, \end{cases}$$

- to find $T_m$, compute a rank-one correction $w$ of $v^m$ and determine the smallest $\alpha$-singular value $\sigma_\alpha$, for each node $\alpha \in T$, of $v^m + w$. Then,

$$T_m = \left\{ \alpha \in T : \sigma_\alpha \geq \theta \max_{\beta \in T} \sigma_\beta \right\},$$

where $\theta \in [0, 1]$. In the next numerical experiments, $\theta$ is set to 0.8.

# Numerical experiment—function with effective dimension 2

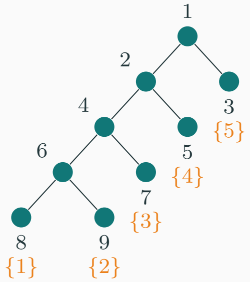
- Approximation in tree-based tensor format of the function $f$ of $X = (X_1, \dots, X_5)$, such that $X_i \sim \mathcal{U}(-1, 1)$:

$$f(X) = \frac{1}{(10 + X_1 + 0.5X_2)^2},$$

- approximation spaces of the leaves: Legendre polynomial basis of maximal degree $5$,
- adaptation of the rank,
- 4 sample sizes: $N = 50$, $100$, $1000$ and $10000$,
- 3 dimension trees.

# Numerical experiment—function with effective dimension 2

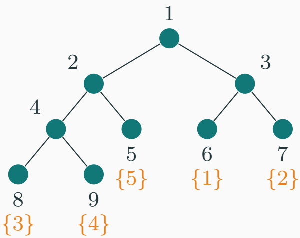
| Format | $N$ | Test error | $r$ |
|---|---|---|---|
| TTT | 50 | $[0.04, 1.46] \cdot 10^{-5}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| | 100 | $[0.22, 2.56] \cdot 10^{-6}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| | 1000 | $[1.45, 1.55] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 1, 1, 3, 3$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| Tree 1 | 50 | $[0.04, 1.80] \cdot 10^{-5}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| | 100 | $[0.21, 8.66] \cdot 10^{-6}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| | 1000 | $[1.45, 1.58] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 3, 3, 1, 1$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| Tree 2 | 50 | $[0.07, 1.69] \cdot 10^{-5}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 100 | $[0.23, 3.60] \cdot 10^{-6}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 1000 | $[1.45, 1.63] \cdot 10^{-7}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |




Tensor-Train Tucker (TTT)

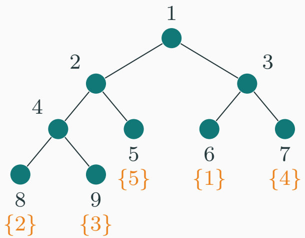
# Numerical experiment—function with effective dimension 2

| Format | $N$ | Test error | $r$ |
|---|---|---|---|
| TTT | 50 | $[0.04, 1.46] \cdot 10^{-5}$ | 1, 1, 1, 1, 1, 1, 1, 4, 4 |
| | 100 | $[0.22, 2.56] \cdot 10^{-6}$ | 1, 1, 1, 1, 1, 1, 1, 4, 4 |
| | 1000 | $[1.45, 1.55] \cdot 10^{-7}$ | 1, 1, 1, 1, 1, 1, 1, 3, 3 |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | 1, 1, 1, 1, 1, 1, 1, 4, 4 |
| Tree 1 | 50 | $[0.04, 1.80] \cdot 10^{-5}$ | 1, 1, 1, 1, 1, 4, 4, 1, 1 |
| | 100 | $[0.21, 8.66] \cdot 10^{-6}$ | 1, 1, 1, 1, 1, 4, 4, 1, 1 |
| | 1000 | $[1.45, 1.58] \cdot 10^{-7}$ | 1, 1, 1, 1, 1, 3, 3, 1, 1 |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | 1, 1, 1, 1, 1, 4, 4, 1, 1 |
| Tree 2 | 50 | $[0.07, 1.69] \cdot 10^{-5}$ | 1, 3, 3, 3, 1, 3, 1, 3, 1 |
| | 100 | $[0.23, 3.60] \cdot 10^{-6}$ | 1, 3, 3, 3, 1, 3, 1, 3, 1 |
| | 1000 | $[1.45, 1.63] \cdot 10^{-7}$ | 1, 3, 3, 3, 1, 3, 1, 3, 1 |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | 1, 3, 3, 3, 1, 3, 1, 3, 1 |




Tree 1

| Format | $N$ | Test error | $r$ |
|---|---|---|---|
| TTT | 50 | $[0.04, 1.46] \cdot 10^{-5}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| | 100 | $[0.22, 2.56] \cdot 10^{-6}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| | 1000 | $[1.45, 1.55] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 1, 1, 3, 3$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 1, 1, 4, 4$ |
| Tree 1 | 50 | $[0.04, 1.80] \cdot 10^{-5}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| | 100 | $[0.21, 8.66] \cdot 10^{-6}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| | 1000 | $[1.45, 1.58] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 3, 3, 1, 1$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 1, 1, 1, 1, 4, 4, 1, 1$ |
| Tree 2 | 50 | $[0.07, 1.69] \cdot 10^{-5}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 100 | $[0.23, 3.60] \cdot 10^{-6}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 1000 | $[1.45, 1.63] \cdot 10^{-7}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |
| | 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | $1, 3, 3, 3, 1, 3, 1, 3, 1$ |




Tree 2

# Numerical experiment—peaked function

- Approximation in tree-based tensor format of the function $f$ of $X = (X_1, \ldots, X_{10})$, such that $X_i \sim \mathcal{U}(0,1)$:

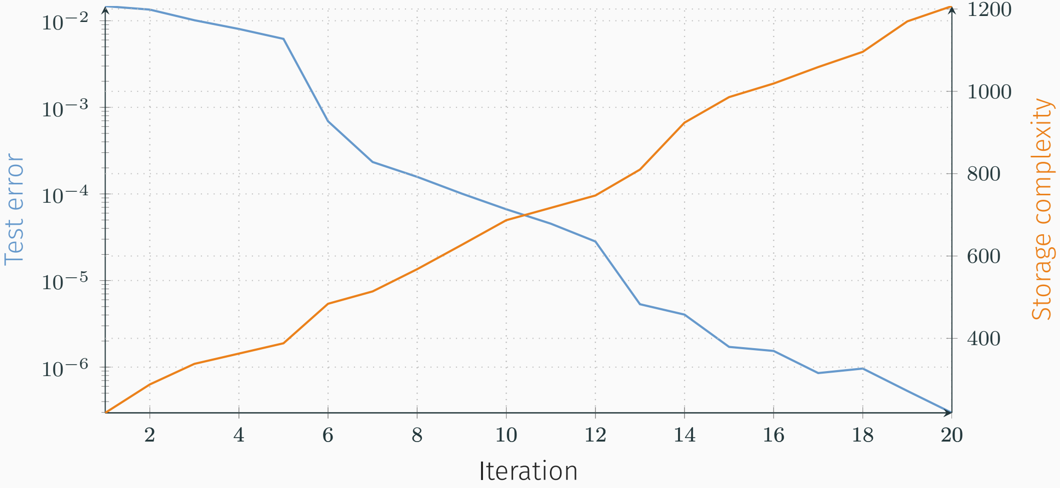
$$f(X) = \left(1 + \sum_{i=1}^{d} a_i x_i\right)^{-2}$$

  with $0 \leq a_i \leq 1$ randomly chosen,
- approximation spaces of the leaves: Legendre polynomial basis of maximal degree $20$,
- adaptation of the rank,
- 3 sample sizes: $N = 100$, $1000$ and $10000$,
- 2 dimension trees: Tensor Train Tucker and Balanced Tree.

# Numerical experiment—peaked function

| Format | $N$ | Test error | $\max_{\alpha \in T} r_\alpha$ | Storage complexity |
|---|---|---|---|---|
| TTT | 100 | $[0.04, 1.94] \cdot 10^{-2}$ | 2 | 425.50 |
| | 1000 | $[0.00, 1.52] \cdot 10^{-2}$ | 3 | 795 |
| | 10 000 | $[0.03, 5.27] \cdot 10^{-5}$ | 3.50 | 817 |
| Balanced tree | 100 | $[0.07, 2.07] \cdot 10^{-2}$ | 2 | 305.50 |
| | 1000 | $[0.00, 1.43] \cdot 10^{-2}$ | 4 | 1019 |
| | 10 000 | $[0.06, 4.68] \cdot 10^{-5}$ | 4 | 1055 |

# Numerical experiment—peaked function—HTT



# Tree-based tensor learning combined with changes of variables

# Tree-based tensor learning combined with changes of variables

## Aim

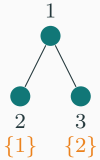
Combine the tree-based tensor learning with changes of variables, by computing a composition of functions:

$$f(x) \approx h(g(x)) = h(g_1(x), \dots, g_m(x)),$$

where $h \in \mathcal{T}_r^T$ with $T \subset 2^{\{1,\dots,m\}}$, $g : \mathbb{R}^d \to \mathbb{R}^m$ and $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, m$.

## Outline of the algorithm

- Sequential construction of tree-based formats with increasing $m$, each time adding a new variable $z_i = g_i(x)$,
- for each $m$, computation of the approximation by alternatively optimizing on
  - $h$, using the method previously introduced,
  - the $g_i$, $i = 1, \dots, m$, using a Gauss-Newton algorithm.



# Tree-based tensor learning combined with changes of variables

## Aim

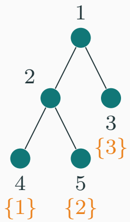
Combine the tree-based tensor learning with changes of variables, by computing a composition of functions:

$$f(x) \approx h(g(x)) = h(g_1(x), \dots, g_m(x)),$$

where $h \in \mathcal{T}_r^T$ with $T \subset 2^{\{1,\dots,m\}}$, $g : \mathbb{R}^d \to \mathbb{R}^m$ and $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, m$.

## Outline of the algorithm

- Sequential construction of tree-based formats with increasing $m$, each time adding a new variable $z_i = g_i(x)$,
- for each $m$, computation of the approximation by alternatively optimizing on
  - $h$, using the method previously introduced,
  - the $g_i$, $i = 1, \dots, m$, using a Gauss-Newton algorithm.



# Tree-based tensor learning combined with changes of variables

## Aim

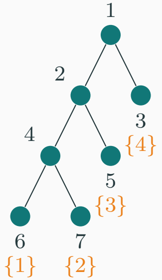
Combine the tree-based tensor learning with changes of variables, by computing a composition of functions:

$$f(x) \approx h(g(x)) = h(g_1(x), \dots, g_m(x)),$$

where $h \in \mathcal{T}_r^T$ with $T \subset 2^{\{1,\dots,m\}}$, $g : \mathbb{R}^d \to \mathbb{R}^m$ and $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, m$.

## Outline of the algorithm

- Sequential construction of tree-based formats with increasing $m$, each time adding a new variable $z_i = g_i(x)$,
- for each $m$, computation of the approximation by alternatively optimizing on
  - $h$, using the method previously introduced,
  - the $g_i$, $i = 1, \dots, m$, using a Gauss-Newton algorithm.



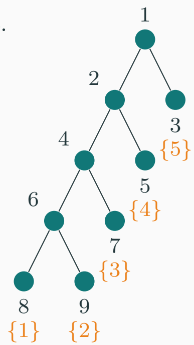
# Tree-based tensor learning combined with changes of variables

### Aim

Combine the tree-based tensor learning with changes of variables, by computing a composition of functions:

$$f(x) \approx h(g(x)) = h(g_1(x), \dots, g_m(x)),$$

where $h \in \mathcal{T}_r^T$ with $T \subset 2^{\{1,\dots,m\}}$, $g : \mathbb{R}^d \to \mathbb{R}^m$ and $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, m$.




### Outline of the algorithm

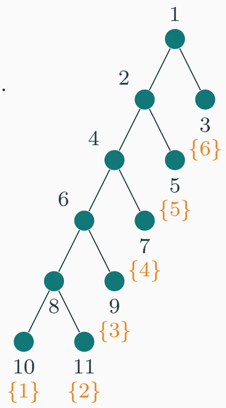
- Sequential construction of tree-based formats with increasing $m$, each time adding a new variable $z_i = g_i(x)$,
- for each $m$, computation of the approximation by alternatively optimizing on
  - $h$, using the method previously introduced,
  - the $g_i$, $i = 1, \dots, m$, using a Gauss-Newton algorithm.

## Aim

Combine the tree-based tensor learning with changes of variables, by computing a composition of functions:

$$f(x) \approx h(g(x)) = h(g_1(x), \dots, g_m(x)),$$

where $h \in \mathcal{T}_r^T$ with $T \subset 2^{\{1,\dots,m\}}$, $g : \mathbb{R}^d \to \mathbb{R}^m$ and $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, \dots, m$.




## Outline of the algorithm

- Sequential construction of tree-based formats with increasing $m$, each time adding a new variable $z_i = g_i(x)$,
- for each $m$, computation of the approximation by alternatively optimizing on
  - $h$, using the method previously introduced,
  - the $g_i$, $i = 1, \dots, m$, using a Gauss-Newton algorithm.

# Numerical experiments—function with effective dimension 2

- Approximation of the previously used function with effective dimension 2, with $X = (X_1, \dots, X_5)$,

$$f(X) = \frac{1}{(10 + X_1 + 0.5X_2)^2},$$

- number of new variables $m$ fixed to 2,
- approximation basis for $g_i$, $i = 1, 2$: multidimensional Legendre polynomial basis of total degree exactly 1 (enables linear combinations of the $X_j$, $j = 1 \dots, d$),
- approximation spaces of the leaves: polynomial basis of maximal degree 5, orthonormal with respect to the measure of $Z = g(X)$.

# Numerical experiments—function with effective dimension 2

## Without changes of variables

| N | Test error | $\max_{\alpha \in T} r_\alpha$ | Storage complexity |
|---|---|---|---|
| 100 | $[0.22, 2.56] \cdot 10^{-6}$ | 4 | 85 |
| 1000 | $[1.45, 1.55] \cdot 10^{-7}$ | 3 | 66 |
| 10 000 | $[1.42, 1.46] \cdot 10^{-7}$ | 4 | 85 |

## With changes of variables (m=2)

| N | Test error | $\max_{\alpha \in T} r_\alpha$ | Storage complexity |
|---|---|---|---|
| 100 | $[0.11, 1.22] \cdot 10^{-5}$ | 2.50 | 46.50 |
| 1000 | $[1.45, 9.85] \cdot 10^{-7}$ | 3 | 55 |
| 10 000 | $[1.40, 5.96] \cdot 10^{-7}$ | 4 | 74 |

# Numerical experiments—Function of dimension 20

- Approximation of the function of dimension 20:

$$f(X) = \sin(\mathrm{w}_2^\intercal X + \mathrm{w}_3^\intercal X)\cos(\mathrm{w}_1^\intercal X + \mathrm{w}_4^\intercal X) + \cos(\mathrm{w}_1^\intercal X + \mathrm{w}_3^\intercal X)$$

  with the $\mathrm{w}_i \in \mathbb{R}^{20}$ taken randomly, $i = 1, \dots, 20$,
- number of new variables $m \leq 4$,
- approximation basis for $g_i$, $i = 1, \dots, m$: multidimensional Legendre polynomial basis of total degree exactly 1,
- approximation spaces of the leaves: polynomial basis of maximal degree 10, orthonormal with respect to the measure of $Z = g(X)$,
- training sample of size 1000.

# Numerical experiments—Function of dimension 20

- Approximation of the function of dimension 20:

$$f(X) = \sin(\mathrm{w}_2^\intercal X + \mathrm{w}_3^\intercal X)\cos(\mathrm{w}_1^\intercal X + \mathrm{w}_4^\intercal X) + \cos(\mathrm{w}_1^\intercal X + \mathrm{w}_3^\intercal X)$$

  with the $\mathrm{w}_i \in \mathbb{R}^{20}$ taken randomly, $i = 1, \dots, 20$,
- number of new variables $m \leq 4$,
- approximation basis for $g_i$, $i = 1, \dots, m$: multidimensional Legendre polynomial basis of total degree exactly 1,
- approximation spaces of the leaves: polynomial basis of maximal degree 10, orthonormal with respect to the measure of $Z = g(X)$,
- training sample of size 1000.

| Change of variables | Test error | $\max_{\alpha \in T} r_\alpha$ | Storage complexity |
|---|---|---|---|
| No | $[1.70, 107\,100] \cdot 10^{-6}$ | 4 | 435 |
| Yes ($m = 4$) | $[1.56, 7.93] \cdot 10^{-3}$ | 5 | 342 |

# Conclusions and outlook

## Conclusion

The statistical learning in tree-based format

- exploits the multilinear representation of tensors by using the classical machinery of linear approximation,
- exploits both low-rank and sparsity,
- can be combined with change of variables techniques.

## Outlook

- Perform tree adaptation,
- combine learning in tree-based format and change of variables using other dimension trees,
- better understand the properties of the approximation set combining tree-based tensor approximation and changes of variables,
- perform a convergence analysis of the proposed algorithm.

# References

Bachmayr, M., R. Schneider, and A. Uschmajew (2016). Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. *Foundations of Computational Mathematics 16*(6), 1423–1472.

Chevreuil, M., R. Lebrun, A. Nouy, and P. Rai (2015). A least-squares method for sparse low rank approximation of multivariate functions. *SIAM/ASA JUQ 3*(1), 897–921.

Cohen, N., O. Sharir, and A. Shashua (2015). On the expressive power of deep learning: A tensor analysis. *CoRR abs/1509.05009*.

Khrulkov, V., A. Novikov, and I. Oseledets (2017). Expressive power of recurrent neural networks.

Lathauwer, L. D., B. D. Moor, and J. Vandewalle (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications 21*(4), 1253–1278.

Nouy, A. (2015). Low-rank tensor methods for model order reduction. In *Handbook of Uncertainty Quantification*, pp. 1–26. Springer International Publishing.

# Thank you for your attention

Contact: `erwan.grelier@ec-nantes.fr`