Adaptive hierarchical low-rank approximation of multivariate functions using statistical methods

Anthony Nouy

Ecole Centrale Nantes

Joint work with Mathilde Chevreuil, Loic Giraldi, Prashant Rai

Supported by the ANR (CHORUS project)

High-dimensional problems in uncertainty quantification

Parameter-dependent models

$$\mathcal{M}(u(X); X) = 0$$

where $X = (X_1, \ldots, X_d)$ are random variables.

- Forward problem: evaluation of statistics, probability of events, sensitivity indices...

$$\mathbb{E}(f(u(X))) = \int_{\mathbb{R}^d} f(u(x))p(x)dx$$

- Inverse problem: from (partial) observations of u, estimate the density of X

$$p(x)$$

- Solving forward and inverse problems requires the evaluation of the model for many instances of X.

High-dimensional problems in uncertainty quantification (UQ)

- In practice, we rely on approximations of the solution map

$$x \mapsto u(x)$$

  which are used as surrogate models.

- Complexity issues:
  - For complex models, only a few evaluations of the function are available.
  - High-dimensional function

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

- Specific low-dimensional structures of functions have to be exploited (low effective dimensionality, anisotropy, sparsity, low rank...)

Outline

Outline

Tensor spaces of multivariate functions

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

The elementary tensor product $v = v^{(1)} \otimes \ldots \otimes v^{(d)}$ of functions $v^{(\nu)} \in V_\nu$ is a multivariate function defined on $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_d$ and such that

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

The algebraic tensor product of spaces V_ν is defined as

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

A Banach tensor space $\overline{V_1 \otimes \ldots \otimes V_d}^{\|\cdot\|}$ is obtained by completion with respect to a norm $\|\cdot\|$.

Here we consider $V_\nu = L^2_{\mu_\nu}(\mathcal{X}_\nu)$ where $\mathcal{X}_\nu$ is equipped with a probability measure μ_ν, and the Hilbert tensor space

$$\overline{L^2_{\mu_1}(\mathcal{X}_1) \otimes \ldots \otimes L^2_{\mu_d}(\mathcal{X}_d)}^{\|\cdot\|} = L^2_\mu(\mathcal{X})$$

with $\mu = \mu_1 \otimes \ldots \otimes \mu_d$ and where $\|\cdot\|$ is the natural norm on $L^2_\mu(\mathcal{X})$.

Canonical rank

The canonical rank of a tensor $v \in V_1 \otimes \ldots \otimes V_d$ is the minimal integer r such that

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

For $d = 2$, it is the unique notion of rank and

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

is a proximinal set and a smooth manifold.

An order-two tensor u in the Hilbert tensor space $\overline{V_1 \otimes V_2}^{\|\cdot\|}$ admits a singular value decomposition

$$u = \sum_{k \geq 1} \sigma_k v_k^{(1)}(x_1) v_k^{(2)}(x_2)$$

An element of best approximation of u from $\mathcal{R}_r$ is given by the truncated singular value decomposition where we retain the r largest singular values.

Canonical rank

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

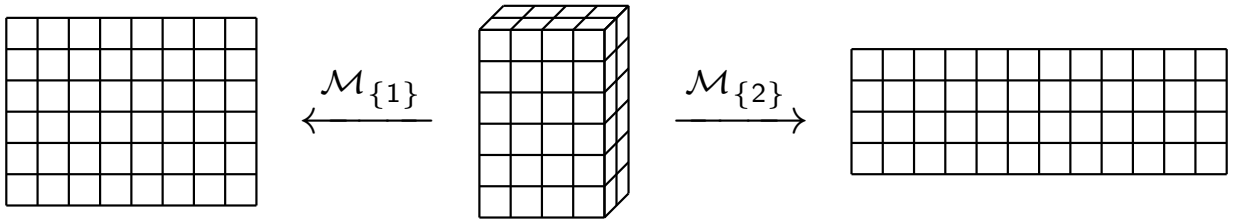
- Determining the rank of a given tensor is a NP-hard problem.
- $\mathcal{R}_r$ is not closed. The consequence is that for most problems involving approximation in canonical format $\mathcal{R}_r$, there is no robust method when $d > 2$.
- The set $\mathcal{R}_r$ is not an algebraic variety.
- No notion of singular value decomposition.

α-rank

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

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

where $V_\alpha = \bigotimes_{\nu \in \alpha} V_\nu$, and $\alpha^c = D \setminus \alpha$.



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

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

which is the minimal integer r_α such that

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

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

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

α-rank

The motivation behind the definition of tensor formats based on α-ranks is to benefit from the nice properties of the two dimensional case.

- The set
$$\mathcal{T}_{r_\alpha}^{\{\alpha\}} = \{v \in V : \text{rank}_\alpha(v) \leq r_\alpha\}$$
of tensors with α-rank bounded by r_α is weakly closed (and therefore proximinal).

- For a given tensor u, $\mathcal{M}_\alpha(u)$ admits a singular value decomposition. A best approximation of u from $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ is provided by the truncated singular value decomposition of $\mathcal{M}_\alpha(u)$ where we retain the r_α largest singular values.

- The determination of the α-rank of a tensor is feasible.

- $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ is a smooth manifold.

α-ranks and related low-rank formats

For T a collection of subsets of D, we define the T-rank of a tensor v, denoted $\text{rank}_T(u)$, as the tuple

$$\text{rank}_T(v) = \{\text{rank}_\alpha(v)\}_{\alpha \in T}.$$

The subset of tensors with T-rank bounded by $r = (r_\alpha)_{\alpha \in T}$ is

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

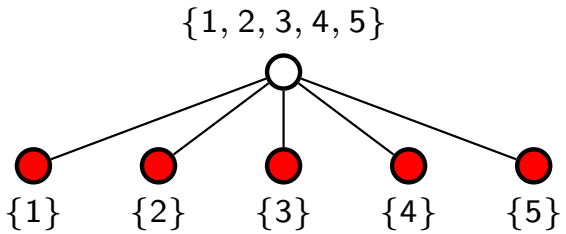
As a finite intersection of subsets $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$, $\mathcal{T}_r^T$ inherits from geometrical and topological properties of the subsets $\mathcal{T}_{r_\alpha}^{\{\alpha\}}$ which are favorable for numerical simulation.

In particular, $\mathcal{T}_r^T$ is weakly closed.

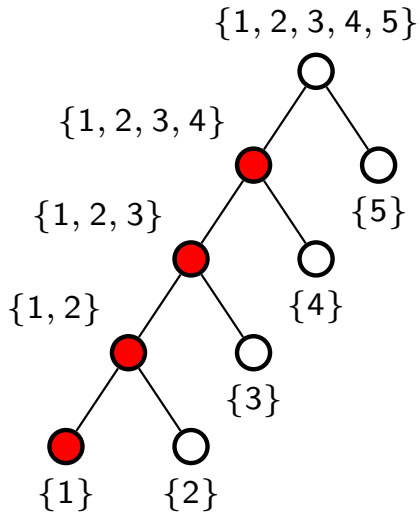
α-ranks and related low-rank formats

Different choices for T yield different tensor formats:

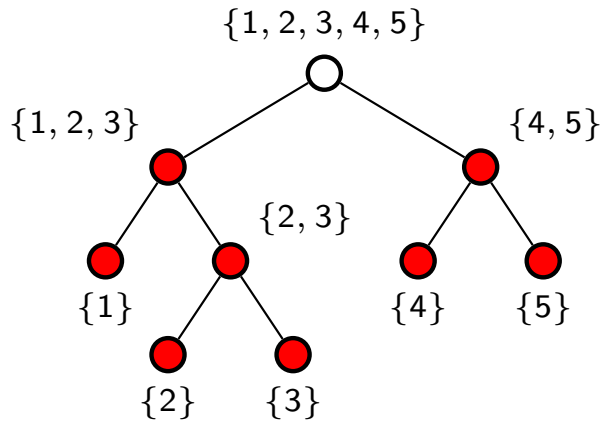
- the Tucker format for $T = \{\{1\}, \ldots, \{d\}\}$
- the Tensor Train format [Oseledets-Tyrtyshnikov'09] for $T = \{\{1\}, \{1,2\}, \ldots, \{1, \ldots, d-1\}\}$
- and more general tree-based (or hierarchical) Tucker formats [Hackbusch-Kuhn'09] for T a tree-structured subset of 2^D.



Tucker



Tensor Train



Tree-based Tucker

Tree-based tensor formats

A tensor $v \in \mathcal{T}_r^T$ admits a representation

$$v(x_1, \ldots, x_d) = \sum_{k_1=1}^{r_1} \cdots \sum_{k_L=1}^{r_L} \prod_{\nu=1}^{d} p^{(\nu)}\left(x_\nu, (k_i)_{i \in s_\nu}\right) \prod_{\nu=d+1}^{M} p^{(\nu)}\left((k_i)_{i \in s_\nu}\right)$$

where the parameter $p^{(\nu)}$ is a tensor which depends on a subset of summation variables $(k_i)_{i \in s_\nu} := k_{s_\nu}$.

- Multilinear parametrization:

$$\mathcal{T}_r^T = \{v = F(p_1, \ldots, p_L); p_k \in P_k, 1 \leq k \leq L\}$$

  where F is a multilinear map.

- Storage complexity scaling as $O(dR^s)$ where $\#S_\nu \leq s$, $r_\nu \leq R$.
- Different extensions of the notion of singular value decomposition for higher-order tensors u, which provide quasi-optimal approximations $u_r \in \mathcal{T}_r^T$ such that

$$\|u - u_r\| \leq \sqrt{\#T} \min_{v \in \mathcal{T}_r^T} \|u - v\|$$

- $\mathcal{T}_r^T$ is a smooth manifold

Outline

Statistical learning methods for tensor approximation

- Approximation of a function $u(X) = u(X_1, \ldots, X_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of X)

Statistical learning methods for tensor approximation

- Approximation of a function $u(X) = u(X_1, \ldots, X_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of X)
- Approximation in subsets of rank-structured functions $\mathcal{M}_r$ by minimization of an empirical risk

$$\widehat{\mathcal{R}}_K(v) = \frac{1}{K} \sum_{k=1}^{K} \ell(u(x^k), v(x^k))$$

where ℓ is a certain loss function.

Statistical learning methods for tensor approximation

- Approximation of a function $u(X) = u(X_1, \ldots, X_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of X)
- Approximation in subsets of rank-structured functions $\mathcal{M}_r$ by minimization of an empirical risk

$$\widehat{\mathcal{R}}_K(v) = \frac{1}{K}\sum_{k=1}^K \ell(u(x^k), v(x^k))$$

where ℓ is a certain loss function.

- Here, we consider for least-squares regression

$$\widehat{\mathcal{R}}_K(v) = \frac{1}{K}\sum_{k=1}^K (u(x^k) - v(x^k))^2 = \widehat{\mathbb{E}}_K((u(X) - v(X))^2)$$

but other loss functions could be used for different objectives than L^2-approximation (e.g. classification).

Alternating minimization algorithm

- Multilinear parametrization of tensor manifolds

$$\mathcal{M}_r = \{v = F(p_1, \ldots, p_L) : p_l \in \mathbb{R}^{m_l}, 1 \leq l \leq L\}$$

so that

$$\min_{v \in \mathcal{M}_r} \widehat{\mathcal{R}}_K(v) = \min_{p_1, \ldots, p_L} \widehat{\mathcal{R}}_K(F(p_1, \ldots, p_L))$$

Alternating minimization algorithm

- Multilinear parametrization of tensor manifolds

$$\mathcal{M}_r = \{v = F(p_1, \ldots, p_L) : p_l \in \mathbb{R}^{m_l}, 1 \leq l \leq L\}$$

so that

$$\min_{v \in \mathcal{M}_r} \widehat{\mathcal{R}}_K(v) = \min_{p_1, \ldots, p_L} \widehat{\mathcal{R}}_K(F(p_1, \ldots, p_L))$$

- Alternating minimization algorithm: Successive minimization problems

$$\min_{p_l \in \mathbb{R}^{m_l}} \widehat{\mathcal{R}}_K(\underbrace{F(p_1, \ldots, p_l, \ldots, p_L)}_{\Psi_l(\cdot)^T p_l})$$

which are standard linear approximation problems

$$\min_{p_l \in \mathbb{R}^{m_l}} \frac{1}{K} \sum_{k=1}^{K} \ell(u(x^k), \Psi_l(x^k)^T p_l)$$

Alternating minimization algorithm

- Regularization

$$\min_{p_I} \frac{1}{K} \sum_{k=1}^{K} \ell(u(x^k), \Psi_I(x^k)^T p_I) + \Omega_I(p_I) \qquad (\star)$$

with regularization functional Ω_I promoting

  - smoothness (of univariate functions),
  - sparsity (e.g. $\Omega_I(p_I) = \lambda_I \|p_I\|_1$ for convex relaxation methods, or a characteristic function for working set algorithms),
  - ...

- $(\star)$ is a standard regularized linear approximation problem.
  - For square-loss and $\Omega_I(p_I) = \lambda_I \|p_I\|_1$, $(\star)$ is a LASSO problem.

- Cross-validation methods for the selection of Ω_I.

Illustrations

- Approximation in tensor-train (TT) format:

$$v(x_1, \ldots, x_d) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} v_{1,i_1}^{(1)}(x_1) v_{i_1,i_2}^{(2)}(x_2) \ldots v_{i_{d-1},1}^{(d)}(x_d)$$

  - Polynomial approximations: $v_{i_{k-1},i_k}^{(k)} \in \mathbb{P}_q$
  - Parametrization: $v = F(p_1, \ldots, p_d)$ with parameter $p_k \in \mathbb{R}^{(q+1)r_k r_{k-1}}$ gathering the coefficients of functions of x_k on a polynomial basis (orthonormal in $L^2_{\mu_k}(\mathcal{X}_k)$).
- Number of parameters:

$$\text{storage}(v) = \sum_{k=1}^{d} r_{k-1} r_k (p+1) = O(d(p+1)R^2)$$

  with $R \geq r_\mu$.
- Sparsity inducing regularization and cross-validation (leave one out) for the automatic selection of polynomial basis functions. Use of standard least-squares in the selected basis.

Illustration : 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)	$U(9855, 12045)$

- Polynomial approximation with degree $q = 8$.
- Test set of size 1000.

Illustration : Borehole function

- Test error for different ranks and for different sizes K of the training set.

rank	K = 100	K=1000	K=10000
(1 1 1 1 1 1 1)	$1.7\,10^{-2}$	$1.4\,10^{-2}$	$1.4\,10^{-2}$
(2 2 2 2 2 2 2)	$6.7\,10^{-4}$	$9.1\,10^{-4}$	$3.3\,10^{-4}$
(3 3 3 3 3 3 3)	$3.2\,10^{-3}$	$1.2\,10^{-4}$	$1.0\,10^{-5}$
(4 4 4 4 4 4 4)	$2.1\,10^{-1}$	$7.6\,10^{-5}$	$1.9\,10^{-7}$
(5 5 5 5 5 5 5)	$7.3\,10^{0}$	$3.8\,10^{-4}$	$2.8\,10^{-7}$
(6 6 6 6 6 6 6)	$7.9\,10^{-1}$	$4.1\,10^{-3}$	$2.1\,10^{-7}$

Illustration : Borehole function

- Test error for different ranks and for different sizes K of the training set.

rank	K = 100	K=1000	K=10000
(1 1 1 1 1 1 1)	$1.7\,10^{-2}$	$1.4\,10^{-2}$	$1.4\,10^{-2}$
(2 2 2 2 2 2 2)	$6.7\,10^{-4}$	$9.1\,10^{-4}$	$3.3\,10^{-4}$
(3 3 3 3 3 3 3)	$3.2\,10^{-3}$	$1.2\,10^{-4}$	$1.0\,10^{-5}$
(4 4 4 4 4 4 4)	$2.1\,10^{-1}$	$7.6\,10^{-5}$	$1.9\,10^{-7}$
(5 5 5 5 5 5 5)	$7.3\,10^{0}$	$3.8\,10^{-4}$	$2.8\,10^{-7}$
(6 6 6 6 6 6 6)	$7.9\,10^{-1}$	$4.1\,10^{-3}$	$2.1\,10^{-7}$

- Finding optimal rank is a combinatorial problem...

Outline

Heuristic strategy for rank adaptation (tree-based Tucker format)

- Given $T \subset 2^{\{1,\dots,d\}}$, construction of a sequence of approximations u_m in tree-based Tucker format with increasing rank:

$$u_m \in \{v : rank_T(v) \leq (r_\alpha^m)_{\alpha \in T}\}$$

Heuristic strategy for rank adaptation (tree-based Tucker format)

- Given $T \subset 2^{\{1,\dots,d\}}$, construction of a sequence of approximations u_m in tree-based Tucker format with increasing rank:

$$u_m \in \{v : rank_T(v) \leq (r_\alpha^m)_{\alpha \in T}\}$$

- At iteration m,

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

where T_m is selected in order to obtain (hopefully) the fastest decrease of the error.

Heuristic strategy for rank adaptation (tree-based Tucker format)

- Given $T \subset 2^{\{1,\dots,d\}}$, construction of a sequence of approximations u_m in tree-based Tucker format with increasing rank:

$$u_m \in \{v : rank_T(v) \leq (r_\alpha^m)_{\alpha \in T}\}$$

- At iteration m,

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

  where T_m is selected in order to obtain (hopefully) the fastest decrease of the error.

- A heuristic strategy consists in computing the singular values

$$\sigma_1^\alpha \geq \dots \geq \sigma_{r_\alpha^m}^\alpha$$

  of α-matricizations $\mathcal{M}_\alpha(u_m)$ of u_m for all $\alpha \in T$

  - $\|u_m\|^2 = \sum_{i=1}^{r_\alpha^m} (\sigma_i^\alpha)^2$ for all $\alpha \in T$.
  - $\sigma_{r_\alpha^m}^\alpha$ provides an estimation of an upper bound of $\|u - u_m\|_{\vee(V_\alpha \otimes V_{\alpha^c})}$
  - Letting $0 \leq \theta \leq 1$, we choose

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

Illustration : Borehole function

- Training set of size $K = 1000$

iteration	rank	test error
0	(1 1 1 1 1 1 1)	$1.4\,10^{-2}$
1	(2 2 2 2 2 2 2)	$4.4\,10^{-4}$
2	(2 2 2 3 3 2 2)	$8.1\,10^{-6}$
3	(3 3 3 4 3 2 2)	$6.2\,10^{-6}$
4	(3 3 3 4 4 3 2)	$2.1\,10^{-5}$
5	(3 3 3 4 4 3 3)	$9.6\,10^{-6}$
6	(3 4 4 4 5 4 4)	$1.5\,10^{-5}$

The selected rank is one order of magnitude better than the optimal "isotropic" rank $(r, r, \ldots, r)$

Illustration : Borehole function

- Training set of size $K = 1000$

iteration	rank	test error
0	(1 1 1 1 1 1 1)	$1.4\,10^{-2}$
1	(2 2 2 2 2 2 2)	$4.4\,10^{-4}$
2	(2 2 2 3 3 2 2)	$8.1\,10^{-6}$
3	(3 3 3 4 3 2 2)	$6.2\,10^{-6}$
4	(3 3 3 4 4 3 2)	$2.1\,10^{-5}$
5	(3 3 3 4 4 3 3)	$9.6\,10^{-6}$
6	(3 4 4 4 5 4 4)	$1.5\,10^{-5}$

The selected rank is one order of magnitude better than the optimal "isotropic" rank $(r, r, \ldots, r)$

- Different sizes K of training set, selection of optimal ranks.

TT format

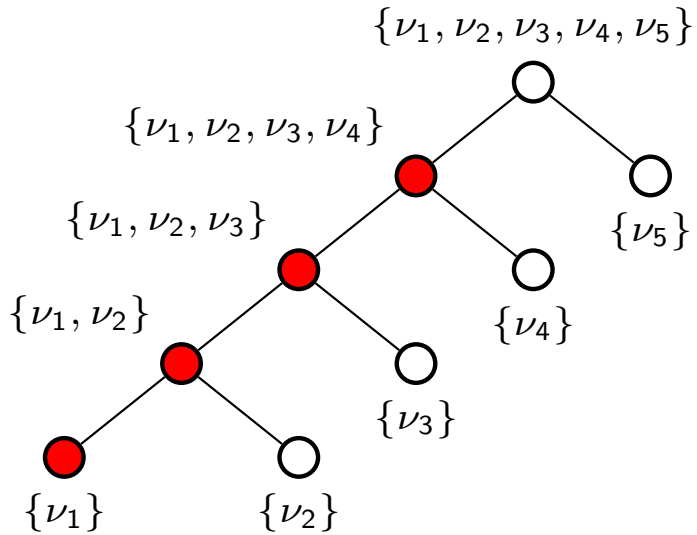
K	rank	test error
100	(3 4 4 3 3 2 1)	$7.1\,10^{-4}$
1000	(3 3 3 4 4 3 2)	$6.2\,10^{-6}$
10000	(5 6 6 7 7 5 4)	$2.4\,10^{-8}$

Canonical format

K	rank	test error
100	2	$1.0\,10^{-3}$
1000	5	$3.8\,10^{-4}$
10000	7	$6.0\,10^{-6}$

Influence of the tree

- Test error for different trees T (Training set of size $K = 50$)



tree	$\{\nu_1, \ldots, \nu_d\}$	optimal rank	test error
T_1	(1 2 3 4 5 6 7 8)	(2 2 2 2 2 1 1)	$6.2\,10^{-4}$
T_2	(1 3 8 5 6 2 4 7)	(2 2 2 2 2 2 1)	$1.3\,10^{-3}$
T_3	(7 6 8 1 4 5 2 3)	(1 1 1 1 1 1 1)	$1.5\,10^{-2}$
T_4	(8 2 4 7 5 1 3 6)	(1 1 2 3 3 2 2)	$1.3\,10^{-2}$

- Finding the optimal tree is a combinatorial problem...

Strategy for tree adaptation

Starting from an initial tree, we perform iteratively the following two steps:

- Run the learning algorithm with rank adaptation to compute an approximation v associated with the current tree

$$v(x_1, \ldots, x_d) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} v_{1,i_1}^{(1)}(x_{\nu_1}) \ldots v_{i_{d-1},1}^{(d)}(x_{\nu_d})$$

- Run a tree optimization algorithm yielding an equivalent representation of v (at the current precision)

$$v(x_1, \ldots, x_d) \approx \tilde{v}(x_1, \ldots, x_d) = \sum_{i_1=1}^{\tilde{r}_1} \cdots \sum_{i_{d-1}=1}^{\tilde{r}_{d-1}} \tilde{v}_{1,i_1}^{(\tilde{\nu}_1)}(x_{\tilde{\nu}_1}) \ldots \tilde{v}_{i_{d-1},1}^{(\tilde{\nu}_d)}(x_{\tilde{\nu}_d})$$

with reduced storage complexity, where $\{\tilde{\nu}_1, \ldots, \nu_d\}$ is a permutation of $\{\nu_1, \ldots, \nu_d\}$.

Strategy for tree adaptation

Illustration with training set of size $K = 50$.
We run the algorithm for different initial trees.
Indicated in blue are the permuted dimensions in the final tree.

tree	$\{\nu_1, \ldots, \nu_d\}$	optimal rank	test error
initial	(1 2 3 4 5 6 7 8)	(2 2 2 2 2 1 1)	$6.2\,10^{-4}$
final	(1 2 3 5 4 6 7 8)	(2 2 2 2 2 1 1)	$4.5\,10^{-4}$
initial	(1 3 8 5 6 2 4 7)	(2 2 2 2 2 2 1)	$1.3\,10^{-3}$
final	(1 3 8 5 2 6 4 7)	(2 2 2 2 2 2 1)	$5.1\,10^{-4}$
initial and final	(7 6 8 1 4 5 2 3)	(1 1 1 1 1 1 1)	$1.5\,10^{-2}$
initial	(8 2 4 7 5 1 3 6)	(1 1 2 3 3 2 2)	$1.3\,10^{-2}$
	(8 2 7 5 1 4 3 6)	(1 1 2 2 2 2 2)	$1.2\,10^{-3}$
final	(8 2 7 5 1 3 4 6)	(1 1 2 2 2 2 2)	$1.3\,10^{-3}$

Concluding remarks

- For rank adaptation, possible use of constructive (greedy) algorithms for tree-based Tucker formats.

Concluding remarks

- For rank adaptation, possible use of constructive (greedy) algorithms for tree-based Tucker formats.
- Need for robust strategies for tree adaptation.

Concluding remarks

- For rank adaptation, possible use of constructive (greedy) algorithms for tree-based Tucker formats.
- Need for robust strategies for tree adaptation.
- "Statistical dimension" of low-rank subsets ?

  For example, the Henon-Heiles potential

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

  has TT-rank $(3,\dots,3)$, a storage complexity scaling as $O(d)$, and the number of samples to recover the function with probability 90% scales as $O(d^{3/2})$.

Concluding remarks

- For rank adaptation, possible use of constructive (greedy) algorithms for tree-based Tucker formats.
- Need for robust strategies for tree adaptation.
- "Statistical dimension" of low-rank subsets ?

  For example, the Henon-Heiles potential

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

  has TT-rank $(3,\dots,3)$, a storage complexity scaling as $O(d)$, and the number of samples to recover the function with probability 90% scales as $O(d^{3/2})$.
- Adaptive/structured sampling strategies.

Concluding remarks

- For rank adaptation, possible use of constructive (greedy) algorithms for tree-based Tucker formats.
- Need for robust strategies for tree adaptation.
- "Statistical dimension" of low-rank subsets ?

  For example, the Henon-Heiles potential

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

  has TT-rank $(3,\dots,3)$, a storage complexity scaling as $O(d)$, and the number of samples to recover the function with probability 90% scales as $O(d^{3/2})$.
- Adaptive/structured sampling strategies.
- Goal-oriented approximations.

References and announcement

A. Nouy.

Low-rank methods for high-dimensional approximation and model order reduction. arXiv:1511.01554

A. Falco, W. Hackbusch, and A. Nouy.

Geometric Structures in Tensor Representations (Final Release).
ArXiv e-prints, May 2015.

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.

M. Chevreuil, L. Giraldi, A. Nouy, and P. Rai.

Learning algorithms for low-rank approximation of multivariate functions in tensor-train tensor format.
Preprint available soon

Open post-doc positions in Ecole Centrale Nantes (France).

Mode order reduction for uncertainty quantification, high-dimensional approximation, low-rank tensor approximation, statistical learning