Hierarchical low-rank approximation for high dimensional approximation

Anthony Nouy

Ecole Centrale Nantes, GeM

Joint work with Mathilde Chevreuil, Loic Giraldi, Prashant Rai

Supported by the ANR (CHORUS project)

Problems in uncertainty quantification

- Parameter-dependent models

$$u : \Xi \to X \quad \text{such that} \quad \mathcal{F}(u(\xi); \xi) = 0$$

where ξ are random variables taking values in a measure space (Ξ, μ).

Problems in uncertainty quantification

- Parameter-dependent models

$$u : \Xi \to X \quad \text{such that} \quad \mathcal{F}(u(\xi); \xi) = 0$$

where ξ are random variables taking values in a measure space (Ξ, μ).

- Forward problem: Given μ, compute a variable of interest

$$s(\xi) = g(u(\xi); \xi)$$

and quantities of interest (e.g. moments, probability of events, sensitivity indices)

Problems in uncertainty quantification

- Parameter-dependent models

$$u : \Xi \to X \quad \text{such that} \quad \mathcal{F}(u(\xi); \xi) = 0$$

where ξ are random variables taking values in a measure space (Ξ, μ).

- Forward problem: Given μ, compute a variable of interest

$$s(\xi) = g(u(\xi); \xi)$$

and quantities of interest (e.g. moments, probability of events, sensitivity indices)

- Inverse problem: Given observations of $s(\xi)$, estimate μ.

Problems in uncertainty quantification

- Classical approach:
  Compute an accurate approximation of $u(\xi)$ or $s(\xi)$ (metamodel, surrogate, reduced order model...) which allows fast evaluations of output variables of interest or observables.

Problems in uncertainty quantification

- Classical approach:
  Compute an accurate approximation of $u(\xi)$ or $s(\xi)$ (metamodel, surrogate, reduced order model...) which allows fast evaluations of output variables of interest or observables.

- **Complexity issues**:
  - Complex numerical models:
  $$u(\xi) \in X, \quad dim(X) = n \gg 1$$
  - Approximation of high-dimensional functions:
  $$s(\xi_1, \dots, \xi_d), \quad d \gg 1$$

High-dimensional approximation

- Approximation of the multivariate function

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

High-dimensional approximation

- Approximation of the multivariate function

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

- "Naive" tensor product discretization yields high-dimensional parametrizations

$$u(x_1, \dots, x_d) \approx \sum_{\alpha_1=1}^{n} \dots \sum_{\alpha_d=1}^{n} a_{\alpha_1 \dots \alpha_d} \psi^{(1)}_{\alpha_1}(x_1) \dots \psi^{(d)}_{\alpha_d}(x_d), \quad a \in \mathbb{R}^{n^d}$$

High-dimensional approximation

- Approximation of the multivariate function

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

- "Naive" tensor product discretization yields high-dimensional parametrizations

$$u(x_1, \dots, x_d) \approx \sum_{\alpha_1=1}^{n} \dots \sum_{\alpha_d=1}^{n} a_{\alpha_1 \dots \alpha_d} \psi^{(1)}_{\alpha_1}(x_1) \dots \psi^{(d)}_{\alpha_d}(x_d), \quad a \in \mathbb{R}^{n^d}$$

  - Number of parameters for full tensor product approximation

$$N = n^d$$

High-dimensional approximation

- Approximation of the multivariate function

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

- "Naive" tensor product discretization yields high-dimensional parametrizations

$$u(x_1, \dots, x_d) \approx \sum_{\alpha_1=1}^{n} \dots \sum_{\alpha_d=1}^{n} a_{\alpha_1 \dots \alpha_d} \psi^{(1)}_{\alpha_1}(x_1) \dots \psi^{(d)}_{\alpha_d}(x_d), \quad a \in \mathbb{R}^{n^d}$$

  - Number of parameters for full tensor product approximation

$$N = n^d$$

  - Assuming $u \in C^s((0,1)^d)$, accuracy

$$\epsilon = O(n^{-s}) = O(N^{-s/d})$$

    and the number of parameters to achieve accuracy ϵ is

$$N = O(\epsilon^{-d/s}) \quad \text{(Curse of dimensionality)}$$

High-dimensional approximation

- Approximation of the multivariate function

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

- "Naive" tensor product discretization yields high-dimensional parametrizations

$$u(x_1, \ldots, x_d) \approx \sum_{\alpha_1=1}^{n} \ldots \sum_{\alpha_d=1}^{n} a_{\alpha_1 \ldots \alpha_d} \psi^{(1)}_{\alpha_1}(x_1) \ldots \psi^{(d)}_{\alpha_d}(x_d), \quad a \in \mathbb{R}^{n^d}$$

  - Number of parameters for full tensor product approximation

$$N = n^d$$

  - Assuming $u \in C^s((0,1)^d)$, accuracy

$$\epsilon = O(n^{-s}) = O(N^{-s/d})$$

    and the number of parameters to achieve accuracy ϵ is

$$N = O(\epsilon^{-d/s}) \quad \text{(Curse of dimensionality)}$$

  - Extra smoothness does not help

Remedies: structured approximations

Specific structures of the function have to be exploited in order to break down the complexity (application dependent)

- Low effective dimensionality,

$$u(x_1, \dots, x_d) \approx u_1(x_1),$$

- Low-order interactions,

$$u(x_1, \dots, x_d) \approx u_0 + \sum_i u_i(x_i)$$

- Sparsity (relatively to a basis or frame)

$$u(x) = \sum_{\alpha \in \mathbb{N}^d} u_\alpha \psi_\alpha(x) \approx \sum_{\alpha \in \Lambda} u_\alpha \psi_\alpha(x)$$

- ...
- Low-rank structures

Outline

1. Rank-structured approximation
2. Statistical learning methods for tensor approximation
3. Adaptive approximation in tree-based low-rank formats

Outline

Rank-structured approximation

- Multivariate function $u(x_1, \ldots, x_d)$ identified with an element of the tensor space

$$V_1 \otimes \ldots \otimes V_d = \overline{span}\{v^1(x_1) \ldots v^d(x_d); v^\mu \in V_\mu\}$$

- Approximation in a subset of tensors with bounded rank

$$\mathcal{M}_{\leq r} = \{v \in V_1 \otimes \ldots \otimes V_d; rank(v) \leq r\}$$

- For order-two tensors, a single notion of rank:

$$rank(v) \leq r \quad \Longleftrightarrow \quad v = \sum_{i=1}^{r} v_i^1(x_1) v_i^2(x_2)$$

- For higher-order tensors, different notions of rank, such as the canonical rank

$$rank(v) \leq r \quad \Longleftrightarrow \quad v = \sum_{i=1}^{r} v_i^1(x_1) \ldots v_i^d(x_d)$$

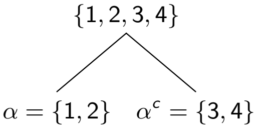
Storage complexity: $O(rdn)$

Subspace based low-rank formats

- α-rank: for $\alpha \subset \{1, \dots, d\}$, $V = V_\alpha \otimes V_{\alpha^c}$, with $V_\alpha = \bigotimes_{\mu \in \alpha} V_\mu$, and

$$rank_\alpha(v) \leq r_\alpha \iff v = \sum_{i=1}^{r_\alpha} v_i^\alpha(x_\alpha) v_i^{\alpha^c}(x_{\alpha^c}), \quad v_i^\alpha \in V_\alpha, \quad v_i^{\alpha^c} \in V_{\alpha^c}$$

$$\iff v \in U_\alpha \otimes V_{\alpha^c} \quad \text{with} \quad \dim(U_\alpha) = r_\alpha$$

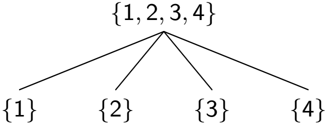


Storage complexity: $O(r_\alpha(n^{\#\alpha} + n^{\#\alpha^c}))$

Subspace based low-rank formats

- Tucker rank:

$$rank_T(v) = (rank_1(v), \ldots, rank_d(v))$$



$$rank_T(v) \leq r = (r_1, \ldots, r_d) \quad \Longleftrightarrow \quad v = \sum_{i_1=1}^{r_1} \ldots \sum_{i_d=1}^{r_d} a_{i_1 \ldots i_d} v_{i_1}^1 \otimes \ldots \otimes v_{i_d}^d$$

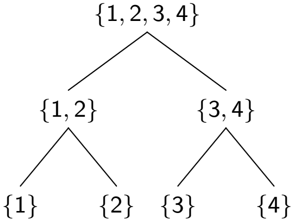
$$\Longleftrightarrow \quad v \in U_1 \otimes \ldots \otimes U_d \quad \text{with} \quad \dim(U_\mu) = r_\mu$$

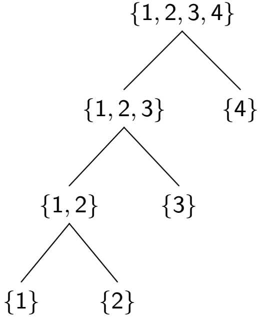
storage complexity: $O(dnR + R^d)$ with $R \geq r_\mu$

Subspace based low-rank formats

- Tree-based Tucker rank [Hackbusch-Kuhn'09,Oseledets-Tyrtyshnikov'09]:

$$rank_T(v) = (rank_\alpha(v) : \alpha \in T) \quad \text{with } T \text{ a dimension tree}$$





storage complexity: $O(dnR + dR^{s+1})$ with $R \geq r_\mu$

Example (additive model): $u(x_1, \dots, x_d) = u_1(x_1) + \dots + u_d(x_d)$ has $rank_T(u) = (2, , 2, \dots, 2)$ for any tree T.

Subspace based low-rank formats

- Storage and computational complexity scales as $O(d)$.
- Best approximation problems in $\mathcal{M}_{\leq r} = \{v : rank_T(v) \leq r\}$ are well-posed, under natural conditions on tensor norms (for reflexive Banach spaces). Follows from the fact that $rank_T(\cdot)$ is weakly l.s.c.. [Falco-Hackbusch-Nouy '15].
- For Hilbert spaces, quasi-optimal approximations with respect to the canonical norm can be obtained by higher-order versions of the SVD

$$\|u - u_r\| \leq \sqrt{2d} \min_{rank_T(v) \leq r} \|u - v\|$$

- The subsets $\mathcal{M}_{\leq r}$ are analytic Banach manifolds and immersed submanifolds under the same conditions on tensor norms [Falco-Hackbusch-Nouy '15] (see [Holtz-Rohwedder-Schneider '11, Uschmajew-Vandereycken '13] for Hilbert setting).
- Multilinear parametrization of subsets of low-rank tensors

$$\mathcal{M}_{\leq r} = \{v = F(p_1, \ldots, p_L); p_k \in P_k, 1 \leq k \leq L\}$$

Outline

Statistical learning methods for tensor approximation

- Approximation of a function $u(\xi) = u(\xi_1, \dots, \xi_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of ξ)

Statistical learning methods for tensor approximation

- Approximation of a function $u(\xi) = u(\xi_1, \dots, \xi_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of ξ)
- Approximation in subsets of rank-structured functions

$$\mathcal{M}_{\leq r} = \{v : rank(v) \leq r\}$$

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 $u(\xi) = u(\xi_1, \ldots, \xi_d)$ from evaluations $\{y_k = u(x^k)\}_{k=1}^K$ on a training set $\{x^k\}_{k=1}^K$ (i.i.d. samples of ξ)
- Approximation in subsets of rank-structured functions

$$\mathcal{M}_{\leq r} = \{v : rank(v) \leq r\}$$

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

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

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

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

so that

$$\min_{v \in \mathcal{M}_{\leq r}} \widehat{\mathcal{R}}_K(v) = \min_{p_1, \dots, p_L} \widehat{\mathcal{R}}_K(F(p_1, \dots, p_d))$$

Alternating minimization algorithm

- Multilinear parametrization of tensor manifolds

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

so that

$$\min_{v \in \mathcal{M}_{\leq r}} \widehat{\mathcal{R}}_K(v) = \min_{p_1, \dots, p_L} \widehat{\mathcal{R}}_K(F(p_1, \dots, p_d))$$

- Alternating minimization algorithm: Successive minimization problems

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

which are standard linear approximation problems

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

Regularization

- Regularization

$$\min_{p_1,\dots,p_L} \widehat{\mathcal{R}}_K(F(p_1,\dots,p_L)) + \sum_{l=1}^{L} \lambda_l \Omega_l(p_l)$$

with regularization functionals Ω_l promoting
  - sparsity (e.g. $\Omega_l(p_l) = \|p_l\|_1$),
  - smoothness,
  - ...
- Alternating minimization algorithm requires the solution of successive standard regularized linear approximation problems

$$\min_{p_l} \frac{1}{K}\sum_{k=1}^{K} \ell(u(x^k), \Psi_l(x^k)^T p_l) + \lambda_l \Omega_l(p_l) \qquad (\star)$$

  - For square-loss and $\Omega_l(p_l) = \|p_l\|_1$, $(\star)$ is a LASSO problem.
- Cross-validation methods for the selection of regularization parameters λ_l.

Illustrations

- Approximation in tensor-train (TT) format:

$$rank_{\{1,\ldots,k\}}(v) \leq r_k, \; \forall k \quad \Longleftrightarrow \quad v(x_1,\ldots,x_d) = \sum_{i_1=1}^{r_1} \ldots \sum_{i_{d-1}=1}^{r_{d-1}} v^1_{1,i_1}(x_1)\ldots v^d_{i_{d-1},1}(x_d)$$

- Polynomial approximations

$$v^k_{i_{k-1},i_k} \in \mathbb{P}_q$$

- $v = F(p_1,\ldots,p_d)$ 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.
- 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:

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

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 $T \subset 2^{\{1,\ldots,d\}}$, 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 $T \subset 2^{\{1,\ldots,d\}}$, 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 $T \subset 2^{\{1,\ldots,d\}}$, 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 possible strategy consists in computing the singular values

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

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

$$\mathcal{M}_\alpha(u_m) = \sum_{i=1}^{r_\alpha^m} \sigma_i^\alpha v_i^\alpha \otimes v_i^{\alpha^c} \in V_\alpha \otimes V_{\alpha^c}$$

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 possible strategy consists in computing the singular values

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

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

$$\mathcal{M}_\alpha(u_m) = \sum_{i=1}^{r_\alpha^m} \sigma_i^\alpha v_i^\alpha \otimes v_i^{\alpha^c} \in V_\alpha \otimes V_{\alpha^c}$$

  - $\sigma_{r_\alpha^m}^\alpha$ provides an estimation of an upper bound of $\|u - u_m\|_{\vee(V_\alpha \otimes V_{\alpha^c})}$

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 possible 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 $\alpha \in T$,

$$\mathcal{M}_\alpha(u_m) = \sum_{i=1}^{r_\alpha^m} \sigma_i^\alpha v_i^\alpha \otimes v_i^{\alpha^c} \in V_\alpha \otimes V_{\alpha^c}$$

  - $\sigma_{r_\alpha^m}^\alpha$ 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

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

TT format

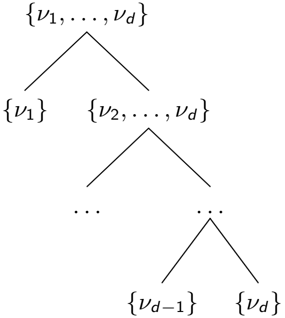
K	rank	test error
100	(3 3 3 4 3 2 2)	$8.4\,10^{-4}$
1000	(3 3 3 4 4 3 2)	$6.2\,10^{-6}$
10000	(5 6 6 7 7 5 4)	$2.4\,10^{-8}$

Canonical format

K	rank	test error
100	3	$6.1\,10^{-4}$
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 $K = 100$)



tree	$\{\nu_1, \dots, \nu_d\}$	optimal rank	test error
T_1	(1 2 3 4 5 6 7 8)	(2 2 2 3 3 2 2)	$1.1\,10^{-4}$
T_2	(7 6 8 1 4 5 2 3)	(1 1 1 1 1 1 1)	$1.4\,10^{-2}$
T_3	(8 2 4 7 5 1 3 6)	(3 3 4 3 3 3 3)	$1.4\,10^{-3}$

Finding the optimal tree is a combinatorial problem...

Concluding remarks and on-going works

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

References and announcement

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

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.

Open post-doc positions in Ecole Centrale Nantes (France).
Mode order reduction for uncertainty quantification, high-dimensional approximation, low-rank tensor approximation, statistical learning