Statistical Learning in Tree-Based Tensor Format

Erwan Grelier†
Centrale Nantes
GeM (UMR CNRS 6183)

Mathilde Chevreuil
Université de Nantes
GeM (UMR CNRS 6183)

Anthony Nouy
Centrale Nantes
LMJL (UMR CNRS 6629)

Statistical learning for uncertainty quantification

We denote by
• \( X = (X_1, \ldots, X_d) \) a set of \( d \) random variables,
• \( Y = f(X) \) the random output of a model \( f \),
   costly to evaluate.

The uncertainty quantification requires many realizations of \( Y \), hence many evaluations of \( f \).
To reduce their cost we use an approximation \( v \) of \( f \), by solving
\[
\min_{v \in \mathcal{V}} \frac{1}{N} \sum_{k=1}^{N} (y_k - v(x_k))^2
\]
with \( \mathcal{V} \) an approximation set and \( (x_k, y_k) \) a realization of \((X, Y)\).

Tree-based (TB) tensor format

A function \( v \in \mathcal{T}_r^T \) in tree-based tensor format associated to a dimension partition tree \( T \subset \mathbb{N}^{1 \ldots d} \)
admits a representation as a composition of multilinear vector-valued functions (tensors):
\[
v(x) = v(x_1, x_2, x_3, x_4) = f_1 f_2 f_3 f_4 g_1 g_2 g_3 g_4
\]
where for the leaves of \( T \), \( 1 \leq \nu \leq d 
\)
\[
f_\nu : \mathbb{R}^{\nu} \rightarrow \mathbb{R}^{\nu},
\]
and for a node \( \alpha \) with \( s \) children \( S(\alpha) \),
\[
f_\alpha : \bigotimes\limits_{\beta \in S(\alpha)} \mathbb{R}^{r_\beta} \rightarrow \mathbb{R}^{r_\alpha}
\]
is a multilinear function identified with a tensor in \( \mathbb{R}^{r_\alpha} \times \mathbb{R}^{r_1} \times \ldots \times \mathbb{R}^{r_s} \).

Properties of the set \( \mathcal{T}_T^r \):
• closed, best approximation problems are well posed,
• storage complexity of \( v \) scaling linearly with \( d \),
• multilinear parametrization of \( v \),
• existence of a higher-order singular value decomposition (HOSVD) of \( v \).

Statistical learning in TB tensor format

Using the multilinear parametrization of \( v \in \mathcal{T}_r^T \), problem (t) is solved using an alternating minimization algorithm, leading to linear problems:
\[
\min_{\alpha \in \mathcal{R}^m} \frac{1}{N} \sum_{k=1}^{N} \| y_k - \Psi_\alpha(x_k) T_{\alpha} \|_2^2, \quad \forall \alpha \in T
\]
with \( \Psi_\alpha(x) \) such that \( v(x) = \Psi_\alpha(x) T_{\alpha} \).

For the leaves of \( T \), a sparse approximation is computed by using a working-set strategy.

Iterative adaptation of \( r \)

Starting from \( r^0 = (0, \ldots, 0) \), at iteration \( n \), given an approximation \( v^n \in \mathcal{T}_r^T \):
• increase the ranks \( (r^k_j)_{k,j} \) associated with a subset of nodes \( T_n \subset T \),
• \( T_n \) is chosen as
\[
T_n = \left\{ \alpha \in T : \sigma_{n, \max} \geq \theta \max_{\beta \in T} \sigma_{\beta} \right\}
\]
with \( \theta \in [0, 1] \) and \( \sigma_n \) the smallest \( \alpha \)-singular value for the node \( \alpha \), obtained by computing the SVD of \( v^n \).

First numerical experiment

Approximation of
\[
f(X) = \sin(w_1^T X + w_2^T X) \cos(w_3^T X + w_4^T X) + \cos(w_5^T X + w_6^T X)
\]
• \( d = 20 \),
• \( X = (X_1, \ldots, X_{10}) \),
• \( X_i \sim \mathcal{U}(-1, 1), i = 1, \ldots, d \),
• \( m \leq 4 \),
• approximation spaces at the leaves: polynomial spaces of maximal degree 10,
• approximation space for the \( g_j, j = 1, \ldots, 4 \) linear functions,
• \( N = 1000 \) training samples.

Second numerical experiment

Approximation of
\[
f(X) = \frac{\sum_{i=1}^{d} a_i X_i}{1 + \sum_{i=1}^{d} a_i X_i}
\]
• \( X = (X_1, \ldots, X_{10}) \), \( X_i \sim \mathcal{U}(0, 1) \),
• \( 0 \leq a_i \leq 1, i = 1, \ldots, d \),
• linear tree (tensor train Tucker format),
• approximation spaces at the leaves: polynomial spaces of maximal degree 20,
• 10000 training samples.

TB learning and changes of variables

An approximation is searched in the form
\[
v(x) = h(g(x)) = h(g_1(x), \ldots, g_m(x)),
\]
with \( h \in \mathcal{T}_r^T \) and \( g_i : \mathbb{R}^d \rightarrow \mathbb{R}, i = 1, \ldots, m \).

Outline of the algorithm

Construction of approximations \( v_{nm} \), at each iteration introducing a new variable \( z_m = g_m(x) \), and alternately optimizing on
• \( h \in \mathcal{T}_r^T \), with fixed \( g \), using the statistical learning algorithm in TB tensor format,
• \( g \), with fixed \( h \), using a Gauss-Newton algorithm,
until convergence.

Confidence interval

Confidence interval

Conclusion

The statistical learning in tree-based tensor format:
• uses the classical machinery of linear approximation,
• exploits both low-rank and sparsity,
• can be combined with changes of variables.

Outlook

• Perform tree adaptation,
• use other trees for \( h \in \mathcal{T}_r^T \) when \( v = h \circ g \),
• characterize the properties of the set of functions of the form \( h \circ g, h \in \mathcal{T}_r^T \).

References