Liberating the dimension for function approximation: Standard information

G.W. Wasilkowski \textsuperscript{a,*}, H. Woźniakowski \textsuperscript{b,c}

\textsuperscript{a} Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA
\textsuperscript{b} Department of Computer Science, Columbia University, New York, NY 10027, USA
\textsuperscript{c} Institute of Applied Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland

\textbf{Abstract}

This is a follow-up paper of “Liberating the dimension for function approximation”, where we studied approximation of infinitely variate functions by algorithms that use linear information consisting of finitely many linear functionals. In this paper, we study similar approximation problems, however, now the algorithms can only use standard information consisting of finitely many function values. We assume that the cost of one function value depends on the number of active variables. We focus on polynomial tractability, and occasionally also study weak tractability. We present non-constructive and constructive results. Non-constructive results are based on known relations between linear and standard information for finitely variate functions, whereas constructive results are based on Smolyak’s construction generalized to the case of infinitely variate functions. Surprisingly, for many cases, the results for standard information are roughly the same as for linear information.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

There is a growing research interest in the complexity and tractability of problems defined on spaces of functions depending on infinitely many variables; see e.g., [1–7]. Most of these papers dealt with integration problems for specific reproducing kernel Hilbert spaces. Algorithms used finitely many function values and the cost of obtaining $f(x)$ was dependent on the number $k(x)$ of “active”
variables in $\mathbf{x}$, and was equal to $S(k(\mathbf{x}))$ for some cost function $S$. Usually, $S$ was linear, whereas in [5], $S$ was an arbitrary monotonically non-decreasing function.

The approximation problem for general Hilbert spaces was studied in [8] with algorithms that used linear information consisting of finitely many linear functionals. The cost of evaluating an inner-product with the representor $h$ was given by $S(|\text{Var}(h)|)$, where $\text{Var}(h)$ denoted the set of indices of all “active” variables in $h$. Similarly to [5], we considered general monotonically non-decreasing $S$.

In [8], we found optimal algorithms and sharp complexity bounds, as well as necessary and sufficient conditions on polynomial and weak tractability. Polynomial tractability means that the complexity of computing an $\varepsilon$-approximation is bounded by a polynomial in $\varepsilon^{-1}$. The smallest such exponent of $\varepsilon^{-1}$ is called the polynomial tractability exponent. Weak tractability means that the complexity is not exponential in $\varepsilon^{-1}$.

The current paper is a continuation of [8] with algorithms restricted to those that only use standard information consisting of finitely many function values. We are mostly interested in polynomial tractability. Weak tractability is only mentioned occasionally. Since algorithms that use arbitrary linear information form a much larger class than algorithms using only standard information, all lower bounds on the complexity for arbitrary linear information also hold for standard information. Moreover, necessary conditions for tractability with arbitrary linear information are also necessary for standard information. Does the same hold for sufficient conditions? Surprisingly, the answer is positive at least for a number of important cases.

We now describe the approach and the results of this paper in a more technical way. Given a separable Hilbert space $H$ of univariate functions, the space $\mathcal{F}$ of infinitely variate functions is obtained as a linear combination of weighted tensor products $H_u$ of $H$. Here $u$ is a finite subset of the set $\{1, 2, \ldots\}$ of natural numbers, and each space $H_u$ contains functions that depend only on the active variables listed in $u$. Then any $f \in \mathcal{F}$ has the unique representation $f = \sum u f_u$, where $f_u \in H_u$ and

$$\|f\|_2^2 = \sum_u \gamma_u^{-1} \cdot \|f_u\|_{H_u}^2 < \infty$$

for a given family of non-negative weights $y = \{\gamma_u\}_u$. The weight $\gamma_u$ monitors the importance of $f_u$. For $\gamma_u = 0$, we assume that $f_u = 0$, and interpret $0/0$ as 0.

Consider next the pre-Hilbert space $\hat{\mathcal{F}}$ of functions from $\mathcal{F}$ with the norm

$$\|f\|_{\hat{\mathcal{F}}}^2 = \sum_u \|f_u\|_{\hat{C}_u}^2,$$

where $\hat{C}_u$ is a weighted $L_2$ space of functions of $|u|$ variables. We need to assume that $\mathcal{F}$ is continuously embedded in $\hat{\mathcal{F}}$. This holds if the weights $\gamma_u$ satisfy the condition (7) stated later. The essence of this condition is that the sequence $\left\{ \sqrt{\gamma_u} \cdot c_{\text{emb}}^{[u]} \right\}_u$ is uniformly bounded, and $c_{\text{emb}}$ is the embedding constant for the univariate case. In particular, we may take $\gamma_u = 1$ for all $u$ only if $c_{\text{emb}} \leq 1$.

By the approximation problem (APP for short) we mean the problem of approximating functions from $\mathcal{F}$ in the norm of $\hat{\mathcal{F}}$, or equivalently, the problem of approximating the embedding operator from $\mathcal{F}$ into $\hat{\mathcal{F}}$.

This problem was studied in [8] for algorithms that use linear information, and it is studied in this paper for algorithms that use standard information. It will be convenient to describe first a few results from [8], and then relate them to the results of this paper.

In [8], we constructed an optimal algorithm $A_{\text{opt}}^\varepsilon$ that is an algorithm whose worst case error is at most $\varepsilon$ and whose cost is minimal. It is interesting that this algorithm is independent of the cost function $\$; however, its cost obviously depends on $\$.

The cost formula of $A_{\text{opt}}^\varepsilon$ gave us an explicit formula for the complexity of the problem, and allowed us to obtain a necessary and sufficient condition for polynomial tractability. These conditions are for general weights and are simplified for product and finite-order weights. Here is the result for general weights.

Let $e_n$ be the minimal worst case error among all algorithms that use at most $n$ linear functionals for the univariate problem, i.e., for the problem restricted to the space $H$ of univariate functions. It is known that $e_n = \sqrt{\lambda_{n+1}}$, where $\{\lambda_n\}_{n \geq 1}$ are ordered eigenvalues of $\text{Emb}^* \circ \text{Emb}$ for the embedding operator $\text{Emb}$ of the univariate case. We have $c_{\text{emb}} = e_0 = \sqrt{\lambda_1}$. Without loss of generality, assume
that \( e_n = \Theta(n^{-p}) \) for some \( p > 0 \), since otherwise the problem cannot be polynomially tractable. Consider
\[
C(\tau) := \sum_u \gamma_u^\tau \cdot \left( \sum_{n=1}^\infty \lambda_n^{\tau n} \right)^{|u|} \quad \text{and} \quad \tau^* := \inf\{ \tau : C(\tau) < \infty \}.
\]
Assume for a moment that the cost function is at most exponential, \( S(d) = \Theta(\exp(k \cdot d)) \). Then
\[
\text{APP is polynomially tractable} \quad \text{iff} \quad \tau^* < \infty \tag{1}
\]
and then the exponent of polynomial tractability equals
\[
p^\text{all} = 2 \cdot \tau^*.
\]
Moreover, if \( \tau^* < \infty \) then APP is weakly tractable even if the cost function is doubly exponential, i.e., \( S(d) = \Theta(\exp(\exp(k \cdot d))) \).

Consider now algorithms that use only standard information. Using the result of [9], we first conclude that we must have
\[
L(1; \lambda) := \sum_{n=1}^\infty \lambda_n < \infty,
\]
since otherwise APP could be intractable for some \( H \), see Proposition 2.

Assume again that \( S(d) = \Theta(\exp(k \cdot d)) \). Then for standard information we have
\[
\text{APP is polynomially tractable} \quad \text{iff} \quad \tau^* < \infty \quad \text{and} \quad L(1; \lambda) < \infty. \tag{2}
\]
If this holds then the exponent of polynomial tractability satisfies
\[
p^\text{std} \leq 2 \cdot (\tau^* + \min(1, [\tau^*]^2)). \tag{3}
\]
The condition (2) is stronger than the condition (1) by the additional requirement that \( L(1; \lambda) < \infty \). This explains the difference between linear and standard information for polynomial tractability. More precisely, for \( e_n = \Theta(n^{-p}) \), we must have \( p > 0 \) for linear information, and at least \( p \geq 1/2 \) for standard information. On the other hand, if \( p > 1/2 \) then \( L(1; \lambda) < \infty \) is automatically satisfied and the conditions (1) and (2) coincide. However, in this case, it is an open problem if the exponents of polynomial tractability might be different for linear and standard information.

We now comment on the exponent \( p^\text{std} \). The proof of the estimate of \( p^\text{std} \) is based on two papers [10,11]. It is an open (and apparently difficult) problem if the corresponding exponents for linear and standard information are the same for all Hilbert spaces of finitely variate functions under the assumption that the exponent for linear information is greater than \( 1/2 \). Any progress on the latter problem would automatically improve the bound (3).

We briefly mention that (2) also implies weak tractability of APP for standard information even when \( S(d) = \Theta(\exp(\exp(k \cdot d^c))) \) for \( c \in (0, 1) \). All these results for standard information can be found in Theorems 3 and 4.

The results mentioned above are non-constructive. This is why we present in Section 4 constructive results by designing algorithms that use standard information and yield polynomial and/or weak tractability. More precisely, suppose that we have algorithms \( A_n \) for the univariate approximation problem with the following properties. Each \( A_n \) uses at most \( n \) function values and its worst case error satisfies
\[
e(A_n; H) = \Theta\left(n^{-1/\kappa}\right) \quad \text{for some} \quad \kappa > 0.
\]
We assume that
\[
\sum_u \gamma_u^{\kappa/2} < \infty \quad \text{and} \quad L(1; \gamma) < \infty. \tag{4}
\]
Using Smolyak’s construction, see [12], and its properties, see [13], we present algorithms \( A^\text{std}_\varepsilon \) for the class \( \mathcal{F} \) whose worst case error and cost satisfy
\[
e(A^\text{std}_\varepsilon; \mathcal{F}) \leq \varepsilon \quad \text{and} \quad \text{cost}(A^\text{std}_\varepsilon) = \Theta\left(S(\varepsilon) \cdot \varepsilon^{-\kappa+o(1)}\right)
\]
with \( d(\varepsilon) = o(\ln(1/\varepsilon)) \). This implies that for \( (d) = \Theta(\exp(k \cdot d)) \),

\[ \text{APP is polynomially tractable with } p^\text{std} \leq \kappa. \]

We stress that for many univariate approximation problems we have \( e(A_n; H) = \Theta(e_n) \). This means that \( A_n \) is optimal to within a factor even among all algorithms using \( n \) linear functionals. For \( e_n = \Theta(n^{-p}) \) we have \( 1/\kappa = p \) and

\[ p^\text{all} = p^\text{std}. \]

In this case, the power of standard and linear information for polynomial tractability is the same. We add that for \( (d) = \Theta(\exp(\exp(k \cdot d))) \), APP is weakly tractable.

Both non-constructive and constructive results are illustrated and extended for special weights such as product and finite-order weights.

In Section 5, we provide an example for which (2) holds, yet (4) does not. This exhibits a trade-off between generality of the assumptions and constructiveness of the results.

Finally, we add that we have chosen \( g \) as the range space of APP since the operator \( \text{APP}^* \circ \text{APP} \) has a relatively simple structure of its spectrum. Using the results of the current papers, we will derive in [14] similar results for an approximation problem with a more interesting/practical range space \( L_2(D, \rho_\infty) \) for which the spectrum of \( \text{APP}^* \circ \text{APP} \) may be very complicated.

2. Basic definitions

We present basic concepts and facts used in this paper. We define quasi-reproducing kernel Hilbert spaces (q-RKHS for short), function approximation, algorithms with their errors and costs, as well as complexity and tractability concepts. We finally recall some of the results from [8] that are needed in this paper.

2.1. Quasi-reproducing kernel Hilbert spaces

We follow here [5], where quasi-reproducing kernel Hilbert spaces have been introduced. Since they are weighted tensor products of spaces of univariate functions, we begin with spaces of functions of one variable.

Consider a separable Hilbert space \( F \) of functions

\[ f : D \rightarrow \mathbb{R}, \]

where \( D \) is a Borel measurable subset of \( \mathbb{R} \). The set \( D \) can be a bounded interval, say \( D = [0, 1] \), or the whole line, \( D = \mathbb{R} \). The norm and inner-product in \( F \) are denoted by \( \| \cdot \|_F \) and \( \langle \cdot, \cdot \rangle_F \). For simplicity of presentation, we assume that \( F \) is of infinite dimension and that

\[ 1 \in F \quad \text{and} \quad \|1\|_F = 1, \]

where \( 1 \) stands for the constant function \( f(x) \equiv 1 \).

Since algorithms studied in the current paper use function evaluations, we have to assume that such evaluations are continuous functionals or, equivalently, that \( F \) is a reproducing kernel Hilbert space (RKHS for short). The reproducing kernel that generates \( F \) is denoted by \( K_F \) and we sometimes write

\[ F = \mathcal{H}(K_F) \]

to stress this fact. Let \( H \) be the subspace of \( F \) that is orthogonal to the constant functions,

\[ H = \{ f \in F : \langle f, 1 \rangle_F = 0 \}. \]

Then \( H \) with the norm

\[ \| \cdot \|_H = \| \cdot \|_F \]

is also a RKHS whose kernel is given by

\[ K_H(x, y) = K_F(x, y) - 1. \]
Clearly, to the norm
\[ K_H(a, a) = 0 \] or equivalently that \( K_F(a, a) = 1 \).
Since \( \|K_H(\cdot, a)\|_F = \sqrt{K_H(a, a)} \), (5) implies that \( K_H(x, a) = 0 \) for all \( x \in D \), and therefore
\[ h(a) = 0 \quad \text{for all} \quad h \in H. \]

We are ready to define the q-RKHS \( \mathcal{F} \) of functions with infinitely (countably) many variables based on the space \( F \) of univariate functions. Let \( D \) be the set of all points \( x \) with infinitely many coordinates,
\[ x = (x_1, x_2, \ldots) \quad \text{with} \quad x_j \in D. \]

Let \( N = \{0, 1, \ldots\} \) and \( N_+ = \{1, 2, \ldots\} \). In what follows, we will use non-empty and finite subsets \( u = \{j_1, j_2, \ldots, j_{|u|}\} \subset N_+ \) with \( 1 \leq j_1 < \cdots < j_{|u|} \) to list active variables \( x_{j_1}, x_{j_2}, \ldots, x_{j_{|u|}} \) among all variables \( x_1, x_2, \ldots \). More precisely, given \( u \), the space \( H_u \) is the RKHS generated by
\[ H_u = \mathcal{H}(K_u) \quad \text{with} \quad K_u(x, y) = \prod_{j \in u} K_H(x_j, y_j). \]

Clearly, any function from \( H_u \) depends only on the active variables \( x_j \) whose indices are listed in \( u \). Note that \( K_u(x, y) = 0 \) if at least one of the active variables of \( x \) or \( y \) is \( a \). Therefore, for all functions \( f \in H_u \), we have
\[ f(x) = 0 \quad \text{if} \quad x_j = a \quad \text{for some} \quad j \in u. \]

Moreover, for any \( f(x) = \prod_{j \in u} f_j(x_j) \) and \( g(x) = \prod_{j \in u} g_j(x_j) \), we have
\[ \langle f, g \rangle_{H_u} = \prod_{j \in u} \langle f_j, g_j \rangle_{H}. \]

For \( u = \emptyset \), we have the space of constant functions,
\[ H_{\emptyset} = \text{span}(1). \]

It has been observed that, in a number of important applications, some groups of variables play more significant roles than other groups, see e.g., the discussion of this point and many references in [15]. To model such spaces, we use a set of weights
\[ \gamma = \{\gamma_u\}_{u \subset N_+, |u| < \infty} \quad \text{with} \quad \gamma_u \geq 0. \]

The weight \( \gamma_u \) quantifies the importance of the group of variables listed in \( u \). In particular, \( \gamma_u = 0 \) means that the corresponding group of variables does not affect the functions from \( \mathcal{F} \). More precisely, let
\[ \Omega_\gamma = \{u \subset N_+ : \gamma_u > 0\}. \]

To avoid a trivial case, we assume that there exists at least one \( u \in \Omega_\gamma \) with \( |u| \geq 1 \).

Let \( H_{\infty, \gamma} \) be the space of all linear combinations of functions from \( \bigoplus_{u \in \Omega_\gamma} H_u \) with the following inner-product
\[ \langle f, g \rangle_{H_{\infty, \gamma}} = \sum_{u \in \Omega_\gamma} \gamma_u^{-1} \cdot \langle f_u, g_u \rangle_{H_u} \quad \text{for} \quad f = \sum_{u \in \Omega_\gamma} f_u \text{ and } g = \sum_{u \in \Omega_\gamma} g_u. \]

Clearly, for \( \gamma_\emptyset > 0 \) we have \( f_\emptyset = f(a) \).

The space \( H_{\infty, \gamma} \) is a pre-Hilbert space. We then define \( \mathcal{F} \) as the completion of \( H_{\infty, \gamma} \) with respect to the norm
\[ \| \cdot \|_{\mathcal{F}} = \| \cdot \|_{H_{\infty, \gamma}}. \]

Clearly, \( \mathcal{F} \) is a separable Hilbert space.
We now check when \( \mathcal{F} \) is a RKHS. If so then for all \( x \in \mathcal{D} \), the linear functional
\[
L_x(f) = f(x) \quad \text{for all } f \in \mathcal{F}
\]
is well defined and continuous. This holds iff
\[
\|L_x\|^2 = \sum_{u \subseteq \mathcal{U}_p} \gamma_u \cdot K_u(x, x) < \infty.
\]
Therefore \( \mathcal{F} \) is a RKHS iff
\[
\sum_{u \subseteq \mathcal{U}_p} \gamma_u \cdot K_u(x, x) < \infty \quad \text{for all } x \in \mathcal{D}. \tag{6}
\]

Then
\[
K(x, y) = \sum_{u \subseteq \mathcal{U}_p} \gamma_u \cdot K_u(x, y) \quad \text{for all } x, y \in \mathcal{D}
\]
is the reproducing kernel of \( \mathcal{F} = \mathcal{H}(\mathcal{K}) \). Note that (6) holds if there are only finitely many positive weights or if
\[
\sum_{u \subseteq \mathcal{U}_p} \gamma_u \cdot \|K_u\|_{\infty} < \infty, \quad \text{where } \|K_u\|_{\infty} := \sup_{x \in \mathcal{D}} K_u(x, x).
\]
If (6) does not hold then \( L_x \) is discontinuous or even ill-defined for some points \( x \in \mathcal{D} \). In this case, we refer to \( \mathcal{F} \) as a quasi-reproducing kernel Hilbert space.

Nevertheless, note that \( L_x \) is always continuous if \( x \) has only finitely many active variables, i.e., all but a finite number of variables of \( x \) are equal to the anchor \( a \). More precisely, for a given point \( x \in \mathcal{D} \) and a subset \( u \), let \((x, u)\) be a short-hand notation for the point \( y = (y_1, y_2, \ldots) \in \mathcal{D} \) with
\[
y_j = \begin{cases} x_j & \text{if } j \in u, \\ a & \text{otherwise.} \end{cases}
\]
Then \( L_{(x, u)}(f) = f(x, u) \) is continuous\(^1\) and
\[
\|L_{(x, u)}\|^2 = \sum_{v \subseteq u} \gamma_v \cdot K_v(x, x) < \infty.
\]

For this reason, the algorithms considered in this paper use only function evaluations at points of the form \((x, u)\).

We end this section with the following important class of q-RKHS.

**Example 1.** Consider first the classical Wiener kernel
\[
K_H(x, y) = \min(x, y) \quad \text{for } x, y \in \mathcal{D} = [0, 1].
\]
The anchor \( a \) is now 0. The space \( H \) consists of absolutely continuous functions with \( f(0) = 0 \) and \( f' \in L_2([0, 1]) \), with the inner-product \( \langle f, g \rangle_H = \int_0^1 f'(x) \cdot g'(x) \, dx \). Since \( \|K_H\|_{\infty} = 1 \), it is easy to see that the condition (6) holds iff
\[
\sum_{u \subseteq \mathcal{U}_p} \gamma_u < \infty.
\]
If so then \( \mathcal{F} \) is a RKHS with the kernel
\[
\mathcal{K}(x, y) = \sum_{u \subseteq \mathcal{U}_p} \gamma_u \cdot \prod_{j \in u} \min(x_j, y_j).
\]

\(^1\) By \( f(x, u) \) we mean \( f((x, u)) \). We drop the extra pair of the parentheses to simplify the notation.
A generalization of this space is provided by taking $D = [T_-, T_+]$ with $T_- \leq 0 < T_+$ or $T_- < 0 \leq T_+$, and

$$
K_H(x, y) = \begin{cases} 
\int_0^{\min(x,y)} \frac{1}{\psi(t)} \, dt & \text{for } x, y \in [0, T_+], \\
\int_0^{\max(x,y)} \frac{1}{\psi(t)} \, dt & \text{for } x, y \in [T_-, 0], \\
0 & \text{otherwise}
\end{cases}
$$

where $\psi$ is a given positive function defined over $D$. We also allow $T_- = -\infty$ or $T_+ = \infty$ so that $D$ can be unbounded. Clearly, the choice of $D = [0, 1]$ and $\psi(t) = 1$ for $t \in [0, 1]$, takes us back to the case discussed before. The role of $\psi$ is to tune the behavior of functions at infinity since for $D = \mathbb{R}$ we have

$$
\|f\|_{L_2}^2 = \int_{-\infty}^{\infty} \frac{|f'(x)|^2}{\psi(x)} \, dx.
$$

For more explanations of the role of $\psi$, the reader is referred to [16], where such spaces have been introduced.

The resulting space $\mathcal{F}$ might only be a q-RKHS. This holds, for example, for $D = [0, 2]$ and $\psi = 1$. We define the weights $\gamma_u = 0$ for all $u \neq \{1, 2, \ldots, k\}$, and $\gamma_{\{1,2,\ldots,k\}} = q^{-k}$ for $q \in [1/2, 1)$ and all $k = 0, 1, \ldots$. Then $K_H(x, y) = \min(x, y)$ for all $x, y \in [0, 2]$. Taking $x = (2, 2, \ldots)$ we obtain

$$
\sum_{u \in \mathcal{U}_y} \gamma_u \cdot K_u(x, x) = \sum_{k=0}^{\infty} (2q)^k = \infty,
$$

showing that $\mathcal{F}$ is not a RKHS.

2.2. Approximation problem

Let $\rho$ be a given probability density function on $D$ and, without loss of generality, assume that $\rho$ is positive over $D$. Then $G = L_2(D, \rho)$ is a Hilbert space of functions $g$ with

$$
\|g\|_G^2 = \int_D |g(x)|^2 \cdot \rho(x) \, dx < \infty.
$$

We assume that $H$ is continuously embedded in $G$. This is equivalent to the condition

$$
C_{\text{emb}} := \sup_{\|f\|_{H} \leq 1} \|f\|_G = \sup_{\|f\|_{H} \leq 1} \left[ \int_D |f(x)|^2 \cdot \rho(x) \, dx \right]^{1/2} < \infty.
$$

For $u \in \mathcal{U}_y$, we let

$$
\rho_u(x) = \prod_{j \in u} \rho(x_j)
$$

and define $\mathcal{G}$ as the space of functions $f = \sum_{u \in \mathcal{U}_y} f_u$ from $\mathcal{F}$ such that

$$
\sum_{u \in \mathcal{U}_y} \|f_u\|_{L_2(D^{\cap u}, \rho_u)}^2 < \infty.
$$

The space $\mathcal{G}$ is the pre-Hilbert space with the inner-product given by

$$
(f, g)_\mathcal{G} := \sum_{u \in \mathcal{U}_y} (f_u, g_u)_{L_2(D^{\cap u}, \rho_u)} \quad \text{for } f = \sum_{u \in \mathcal{U}_y} f_u \text{ and } g = \sum_{u \in \mathcal{U}_y} g_u.
$$
Then the embedding operator
\[ S : \mathcal{F} \to \mathcal{G} \] such that \( S(f) = f \)
is continuous iff
\[
\| S \| = \sup_{u \in U} \sqrt{u_{\gamma}} \cdot C_{\text{emb}}^{\|u\|} < \infty.
\] (7)

This is why we assume throughout this paper that (7) holds.

We are interested in approximating \( S(f) \) for \( f \in \mathcal{F} \) with error measured in \( \mathcal{G} \). We will refer to this problem as the function approximation problem (APP for short).

2.3. Algorithms

Functions \( f \in \mathcal{F} \) are approximated by algorithms that, without loss of generality,\(^2\) have the following form
\[
A_n(f) = \sum_{i=1}^{n} f(x_i, u_i) \cdot a_i,
\]
where \( a_i \in \mathcal{F} \) and \((x_i, u_i)\) are the sampling points with finitely many active variables. We stress that \( A_n \) is well defined and continuous.

The worst case error of \( A_n \) is defined by
\[
e(A_n; \mathcal{F}) := \sup_{\|f\| \leq 1} \|f - A_n(f)\|_G = \|S - A_n\|,
\]
and it is equal to the operator norm of \( S - A_n \).

Following [5,8], we define the information cost (or simply cost) of \( A_n \) by
\[
\text{cost}(A) := \sum_{i=1}^{n} S(|u_i|)
\]
for a given cost function \( S \),
\[
S : [0, \infty) \to [1, \infty).
\]

We consider quite a general class of cost functions by assuming only that \( S \) is monotonically non-decreasing. In another words, the cost of the algorithm \( A_n \) is the sum of the costs \( S(|u_i|) \) of evaluating \( f \) at points \((x_i, u_i)\) with \(|u_i|\) active variables.

2.4. Complexity and tractability

Let \( \varepsilon \in (0, 1) \) be an error demand. The worst case \( \varepsilon \)-information complexity (or \( \varepsilon \)-complexity for short) is the smallest cost among all algorithms \( A_n \) with \( e(A_n; \mathcal{F}) \leq \varepsilon \),
\[
\text{comp}(\varepsilon; \mathcal{F}) := \inf \{ \text{cost}(A_n) : A_n \text{ such that } e(A_n; \mathcal{F}) \leq \varepsilon \}.
\]

We mostly study polynomial tractability in this paper. APP is polynomially tractable iff there are non-negative numbers \( p \) and \( C \) such that
\[
\text{comp}(\varepsilon; \mathcal{F}) \leq C \cdot \varepsilon^{-p} \text{ for all } \varepsilon > 0.
\] (8)
The polynomial tractability exponent \( p^{\text{std}} \) is defined as the infimum of \( p \) satisfying (8), i.e.,
\[
p^{\text{std}} = \limsup_{\varepsilon \to 0} \frac{\text{comp}(\varepsilon; \mathcal{F})}{\ln(1/\varepsilon)}.
\]

\(^2\) It is well known that for linear problems, such as APP, nonlinear algorithms and adaption do not help, see, e.g., [17].
Occasionally, we also discuss weak tractability. APP is weakly tractable iff
\[
\limsup_{\epsilon \to 0} \epsilon \cdot \ln(\text{comp}(\epsilon; F)) = 0,
\]
otherwise we say that APP is intractable.

We want to add that there are different types of tractability such as quasi-polynomial or \( T \)-tractability but we do not address them in this paper.

2.5. Unrestricted linear information

As already mentioned, we study in this paper algorithms using only standard information that consists of values of \( f \) at points \((x_i, u_i)\). Algorithms using arbitrary linear information were considered in [8]. Since we will need some of the results from that paper, we recall them briefly.

The algorithms considered in [8] take the form
\[
A_n(f) = \sum_{i=1}^{n} (f, h_i)_F \cdot a_i,
\]
where \( h_i \) are functions from \( F \). The worst case error of such algorithms is defined, as in this paper, by the operator norm \( \| S - A_n \| \), and the cost is defined as follows. Given
\[
h = \sum_{u \in \mathcal{U}} h_u \quad \text{with} \quad h_u \in H_u,
\]
let
\[
\text{Var}(h) := \{ u : h_u \neq 0 \}
\]
be the set of indices of all active variables in \( h \). Then the cost of evaluating the inner-product \( (f, h)_F \) is given by
\[
\text{cost}(h) := $(|\text{Var}(h)|),
\]
and the cost of \( A_n \) is
\[
\text{cost}(A_n) := \sum_{i=1}^{n} \text{cost}(h_i).
\]

Clearly, algorithms using only standard information are examples of algorithms given by (9) with
\[
f(x, u) = (f, h)_F \quad \text{for} \quad h = \sum_{v \in \mathcal{U}^t, v \subseteq u} \gamma_v \cdot K_v(\cdot, (x, v))
\]
and \( \text{Var}(h) = \bigcup_{v \in \mathcal{U}^t, v \subseteq u} v \subseteq u \). Therefore
\[
\text{cost}(h) \leq $(|u|).
\]

Obviously, if \( u \in \mathcal{U}^t \) then \( \text{cost}(h) = $(|u|) \).

Once the error and cost are defined, the complexity of APP and its tractability are defined as before. We stress the use of arbitrary linear information by adding ‘All’ or ‘all’ in the notation. In particular, \( \text{comp}(\epsilon; F, \text{All}) \) and \( p^{\text{all}} \) denote the \( \epsilon \)-complexity and the polynomial tractability exponent, respectively, when arbitrary linear information is allowed. Of course,
\[
\text{comp}(\epsilon; F, \text{All}) \leq \text{comp}(\epsilon; F) \quad \text{and} \quad p^{\text{all}} \leq p^{\text{std}}.
\]

Let \( S : H \to G \) be the embedding operator given by \( S(f) = f \). As shown in [8], the complexity of APP depends on the weights \( \mathcal{Y} \) as well as on the eigenvalues of the following operator \( W \):
\[
W := S^* \circ S : H \to H.
\]
It is self-adjoint and positive definite. Moreover,

\[ W(f)(x) = \int_D f(t) \cdot K_H(x, t) \cdot \rho(t) \, dt, \]

see, e.g., [17]. To have a finite complexity (even for the univariate case), we have to assume that \( W \) is compact. Let \( \{(\lambda_j, \eta_j)\}_{j \in \mathbb{N}^+} \) be the eigenpairs of the compact operator \( W \). Without loss of generality, we assume that

\[ \lambda_j \geq \lambda_{j+1} \quad \text{and} \quad \langle \eta_j, \eta_k \rangle_W = \delta_{j,k}. \]

Since \( \text{dim}(H) = \infty \), all eigenvalues \( \lambda_j \) are positive.

Let \( e^{\text{all}}(n, H) \) denote the minimal worst case error of approximating univariate functions \( f \) from \( H \) by algorithms using at most \( n \) continuous linear functionals,

\[ e^{\text{all}}(n; H) = \inf_{L_n \in \mathcal{L}^n, \eta_n \in H} \sup_{\|f\|_H \leq 1} \left\| f - \sum_{i=1}^n L_i(f) \cdot a_i \right\|_G. \]

It is well known, see e.g., [17], that

\[ e^{\text{all}}(n; H) = \inf_{L_n \in \mathcal{L}^n, f, L_n(f) = 0, \|f\|_H \leq 1} \|f\|_G = \sqrt{\lambda_{n+1}} \quad \text{for all } n \in \mathbb{N}. \]

The functions \( \{\eta_j\}_{j \geq 1} \) form a complete orthonormal system of \( H \). Letting \( \eta_0 = 1 \), the functions \( \{\eta_j\}_{j \geq 0} \) form a complete orthonormal system of \( F \).

For a non-empty subset \( u = \{u_1, u_2, \ldots, u_d\} \) with \( u_1 < u_2 < \cdots < u_d \), \( d = |u| \), and \( j = (j_1, j_2, \ldots, j_d) \in \mathbb{N}_+^d \), define

\[ \eta_{j,u}(x) := \prod_{k=1}^d \eta_{j_k}(x_{u_k}) \quad \text{and} \quad \lambda_{j,u} := \prod_{k=1}^d \lambda_{j_k}. \]

Then the functions \( \{\eta_{j,u}\}_{j \in \mathbb{N}_+^d} \) form a complete orthonormal system in \( H_u \) and

\[ W_u(\eta_{j,u}) = \lambda_{j,u} \cdot \eta_{j,u}, \]

where \( W_u \) is the \( |u| \)-fold tensor product of \( W \), i.e.,

\[ W_u = S_{\eta_1}^* \circ S_u : H_u \to H_u \quad \text{and} \quad W_u(f)(x) = \prod_{k=1}^d W(f_k)(x_{u_k}) \text{ for } f(t) = \prod_{k=1}^d f_k(t_{u_k}). \]

We now have \( C_{\text{emb}} = \sqrt{\lambda_1} \) and the condition (7) takes the form

\[ \|\delta\| = \sup_{u \in \mathcal{L}_Y} (\gamma_u \cdot \lambda_1)^{1/2} < \infty. \]

For a given error demand \( \varepsilon \), define

\[ M(\varepsilon, u) := \left\{ j \in \mathbb{N}^{|u|} : \gamma_u \cdot \prod_{k=1}^{|u|} \lambda_{j_k} > \varepsilon^2 \right\}, \]

as well as

\[ \mathcal{U}(\varepsilon) := \left\{ u \in \mathcal{L}_Y : \gamma_u \cdot \lambda_1^{|u|} > \varepsilon^2 \right\} \quad \text{and} \quad d(\varepsilon) := \sup\{|u| : u \in \mathcal{U}(\varepsilon)\}. \quad (10) \]

As shown in [8, Thm. 1], the following algorithm has the smallest cost among all algorithms using linear information with errors bounded by \( \varepsilon \):

\[ A^{\text{opt}}_\varepsilon(f) := \sum_{u \in \mathcal{U}(\varepsilon)} A^{\text{opt}}_{\varepsilon,u}(f) \quad \text{with} \quad A^{\text{opt}}_{\varepsilon,u}(f) = \sum_{j \in M(\varepsilon, u)} \langle f, \eta_{j,u} \rangle_{\mathcal{F}} \cdot \eta_{j,u}. \quad (11) \]
That is,
\[ e(\mathcal{A}_{\varepsilon}^{\text{opt}}; F) \leq \varepsilon \quad \text{and} \quad \text{comp}(\varepsilon; F, \Lambda^{\text{all}}) = \text{cost}(\mathcal{A}_{\varepsilon}^{\text{opt}}). \quad (12) \]

Note that the algorithm \( \mathcal{A}_{\varepsilon}^{\text{opt}} \) does not depend on the cost function \$, however, its cost and the complexity obviously depend on \$.

Due to orthogonality of subspaces \( H_{\varepsilon} \), we have
\[ A_{\varepsilon,u}(f) = A_{\text{opt},u}(f_{\varepsilon}). \]
Moreover, for \( f \in H_{\varepsilon} \) with \( u \) outside of the set \( \Omega(\varepsilon) \), \( A_{\varepsilon,u}(f) = 0 \) since \( \| \delta(f) \|_{\gamma} \) does not exceed \( \varepsilon \) for such functions.

The proofsof our results for standard information will usethesesfacts.

Due to (12), the study of complexity and/or tractability of APP for linear information is equivalent to the analysis of the cost of \( \mathcal{A}_{\varepsilon}^{\text{opt}} \). We now recall some of the results of [8] pertaining totractability conditions when \( \Lambda^{\text{all}} \) is allowed. These conditions will be also needed for the tractability study with standard information.

Let \( \lambda = \{\lambda_j\}_{j \geq 1} \) denote the sequence of the eigenvalues of \( W \). Then
\[ \text{ decay} \lambda := \sup\{p \geq 0 : \lim_{j \to \infty} \lambda_j \cdot j^p = 0\} \]
denotes the polynomial decay of eigenvalues. Since \( \lim_{j \to \infty} \lambda_j = 0 \), the decay is well defined and is non-negative. For \( \tau > 0 \), define
\[ C(\tau) := \sum_{u \in \Omega_{\gamma}} \gamma_u^{\tau} \cdot [L(\tau; \lambda)]^{\|u\|} \quad \text{with} \quad L(\tau; \lambda) := \sum_{j=1}^{\infty} \lambda_j^{\tau}. \]
Note that \( C(\tau) < \infty \) implies that \( L(\tau; \lambda) < \infty \) since at least one \( \gamma_u \) is positive with \( |u| \geq 1 \). The eigenvalues \( \lambda_j \) are ordered, and therefore
\[ \lambda_j \leq j^{-1/\tau} \cdot [L(\tau, \lambda)]^{1/\tau}. \]
This means that \( C(\tau) < \infty \) implies \( \tau \geq 1/\text{decay} \lambda \). Note also that \( L(\tau; \lambda) < \infty \) for \( \tau > 1/\text{decay} \lambda \). For \( \tau = 1/\text{decay} \lambda \) we may have \( L(\tau; \lambda) \) finite or infinite. Indeed, take
\[ \lambda_j = \frac{1}{j^\alpha \ln^\beta(j + 1)} \quad \text{for all} \quad j \in \mathbb{N}, \]
for \( \alpha > 0 \) and \( \beta \geq 0 \). Then \( \text{ decay} \lambda = \alpha \) and \( L(1/\alpha; \lambda) < \infty \) iff \( \beta > \alpha \).

From Jensen’s inequality we have
\[ C(\tau) \leq C(c \cdot \tau)^{1/c} \quad \text{for all} \quad c \in (0, 1]. \]
Therefore
\[ C(\tau) < \infty \quad \longrightarrow \quad C(\tau_1) < \infty \quad \text{for all} \quad \tau_1 > \tau. \]

Define
\[ \tau^* = \tau^*(\gamma, \lambda) := \inf\{\tau > 0 : C(\tau) < \infty\} \]
with the convention that the infimum of the empty set equals to infinity. Hence, \( \tau^* < \infty \) iff \( C(\tau) < \infty \) for some \( \tau \), and this \( \tau \) must be at least \( 1/\text{decay} \lambda \). Therefore
\[ \tau^*(\gamma, \lambda) \geq \frac{1}{\text{decay} \lambda}. \]

We are ready to recall conditions on polynomial tractability for the class \( \mathcal{A}_{\text{all}}^{\text{all}} \) with the simplifying assumption that \( \lim \sup_{d \to \infty} \ln(S(d))/d < \infty \), i.e., \( S(d) = e^{o(d)} \). From [8, Thm. 4] we know that
\[ \text{APP is polynomially tractable} \quad \iff \quad \tau^*(\gamma, \lambda) < \infty. \quad (13) \]
Furthermore, the polynomial tractability exponent is given by \( p^\text{all} = 2 \cdot \tau^*(\gamma, \lambda) \), and
\[
\lim_{\varepsilon \to 0} \frac{d(\varepsilon)}{\ln(1/\varepsilon)} = 0.
\]
Finally, for any positive \( \tau \) with finite \( C(\tau) \), we have
\[
\text{cost}(A^\text{opt}_\varepsilon) \leq C(\tau) \cdot \varepsilon^{-2\tau} \cdot \$(d(\varepsilon)),
\]
where \( d(\varepsilon) \) is given by (10).

The significance of (14) is that we can restrict ourselves to algorithms that use function samplings at points with at most \( d(\varepsilon) = o\left(\ln\left(\frac{1}{\varepsilon}\right)\right) \) active variables. This means that the cost of each sampling will never exceed \( \$(o\left(\ln\left(\frac{1}{\varepsilon}\right)\right)) \).

Since the condition \( C(\tau) < \infty \) is essential, we end this section by providing a simplified upper bound on \( C(\tau) \) for an important class of weights, the so-called \textit{product weights}. Recall that \( \gamma = \{\gamma_u\}_u \) is a set of product weights if
\[
\gamma_u = \prod_{k=1}^{1} \gamma_{u_k}
\]
for a given sequence of non-negative numbers \( \{\gamma_j\}_{j \geq 1} \). Then
\[
C(\tau) = \prod_{j=1}^{\infty} (1 + \gamma_j^\tau L(\tau; \lambda)).
\]
Let \( L(\tau; \gamma) = \sum_{k=1}^{\infty} \gamma_k^\tau \). We have
\[
C(\tau) < \infty \quad \text{iff} \quad L(\tau; \lambda) < \infty \quad \text{and} \quad L(\tau; \gamma) < \infty,
\]
and then
\[
C(\tau) \leq e^{L(\tau; \lambda) - L(\tau; \gamma)}.
\]
Hence
\[
C(\tau) < \infty \quad \text{if} \quad \tau > \max\left(\frac{1}{\text{decay}_\lambda}, \frac{1}{\text{decay}_\gamma}\right).
\]
Moreover,
\[
\tau^*(\gamma, \lambda) = \max\left(\frac{1}{\text{decay}_\lambda}, \frac{1}{\text{decay}_\gamma}\right).
\]

3. Non-constructive results

In this section, we present upper bounds on the complexity as well as tractability results for standard information using non-constructive arguments. Constructive arguments will be used in the next section; however, they are based on a different type of assumptions which, sometimes, are more restrictive than for non-constructive results as shown in Section 5.

We want to obtain complexity and tractability results based only on the properties of the univariate eigenvalues \( \lambda \) and the weights \( \gamma \), and which are independent of the structure of the Hilbert space \( H \). Recall that
\[
L(1; \lambda) = \sum_{j=1}^{\infty} \lambda_j = \int_D K_H(x, x) \cdot \rho(x) dx = \text{trace}(W).
\]
We first observe that \( L(1; \lambda) < \infty \) is a necessary condition on tractability for general spaces \( H \). This result easily follows from [9].

\textbf{Proposition 2.} There exist a separable RKHS \( H \) of univariate functions and a probability density function \( \rho \) for which the operator \( W \) is compact, \( L(1; \lambda) = \infty \), and the corresponding APP problem over \( \mathcal{F} \) is intractable for any weights \( \gamma \) with at least one \( \gamma_u > 0 \) for \( |u| \geq 1 \).
Similarly to \( e^{\text{all}}(n; H) \), let \( e^{\text{std}}(n; H) \) denote the \( n \)th minimal error when only standard information is used, i.e.,

\[
e^{\text{std}}(n; H) := \inf_{x_i \in D, a_i \in H} \sup_{\|f\|_H \leq 1} \left\| f - \sum_{i=1}^{n} f(x_i) \cdot a_i \right\|_G = \inf_{x_i \in D} \sup_{\|f(x)\|_H \leq 1} \|f\|_G.
\]

It was shown in [9] that there is a separable RKHS \( H \) and a weight \( \rho \) for which the corresponding eigenvalues \( \lambda_j \) satisfy \( L(1; \lambda) = \infty \), \( \lim_{n \to \infty} \lambda_n = 0 \), and

\[
e^{\text{std}}(n; H) \geq \frac{1}{\ln(\ln(n))} \quad \text{for some subsequence of } n \text{ tending to infinity.}
\]

Consider now the corresponding \( F \) and APP problem. Let \( A_n \) be an arbitrary linear algorithm using function values at \( (x_i, u_i) \) for \( i = 1, 2, \ldots, n \).

Let \( \delta \in (0, 1) \). Take any non-empty \( u^* = \{u_1^*, \ldots, u_k^*\} \) for which \( \gamma_{u^*} > 0 \), and a function \( w \in H_{u^*} \) of the form

\[
u(y) = \gamma_u^{1/2} \cdot \eta_1(y_{u_1^*}) \cdot \eta_1(y_{u_2^*}) \cdots \eta_1(y_{u_{k-1}^*}) \cdot g(y_{u_k^*}).\]

The function \( \eta_1 \) is the normalized eigenfunction of \( W \) corresponding to the largest eigenvalue \( \lambda_1 = C^2_{\text{emb}} \). The function \( g \in H \) satisfies \( \|g\|_H = 1 \), \( \|g\|_C \geq e^{\text{std}}(n, H) \cdot (1 - \delta) \), and \( g(x_{i,k}) = 0 \) for all \( i = 1, 2, \ldots, n \), where \( x_{i,k} \) denotes the \( u_i^* \)th variable of the sampling point \( (x_i, u_i) \). Then \( w((x_i, u_i)) = 0 \) for all \( i = 1, 2, \ldots, n \), \( \|w\|_F = 1 \), and \( A_n(w) = 0 \). Hence

\[
e(A_n; F) \geq \|w\|_g \geq \gamma_u^{1/2} \cdot C^{k-1}_{\text{emb}} \cdot \frac{1 - \delta}{\ln(\ln(n))}
\]

for some subsequence of \( n \) tending to infinity. This obviously implies intractability of APP, as claimed. \( \square \)

Since we are dealing with general RKHS \( H \), finite \( L(1; \lambda) \) is therefore necessary for weak tractability. This is why we will assume throughout the rest of the paper that

\[
L(1; \lambda) < \infty.
\]

(15)

Obviously, \( (15) \) implies that

\[
decay_\lambda \geq 1.\]

This should be contrasted with \( (13) \). For polynomial tractability in the class \( A^{\text{all}} \) (with \( \lim \sup_{n} \ln(S(d))/d < \infty \)) we only need to assume that \( \text{decay}_\lambda > 0 \), whereas for the class \( A^{\text{std}} \), independently of the cost function \( S \), we need to assume that \( \text{decay}_\lambda \geq 1 \). For the class \( A^{\text{all}} \) we also need to assume that \( \tau^*(\gamma, \lambda) < \infty \). As we shall see, this assumption remains sufficient for the class \( A^{\text{std}} \).

We consider two cases: \( \tau^*(\gamma, \lambda) < 1 \), and \( \tau^*(\gamma, \lambda) \geq 1 \). Note that \( \tau^*(\gamma, \lambda) < 1 \) implies that \( \text{decay}_\lambda > 1 \) and \( L(1; \lambda) < \infty \); whereas for \( \tau^*(\gamma, \lambda) \geq 1 \) we only know that \( \text{decay}_\lambda \geq 1 \) due to \( (15) \).

3.1. Case 1: \( \tau^*(\gamma, \lambda) < 1 \)

We now use results from [10] which we state in a slightly different way than it was done in [10, Thm. 5]. For \( d \in \mathbb{N}_+ \), let \( F_d = H(K_d) \) be a separable RKHS of functions with the domain \( D_d \subseteq \mathbb{R}^d \), and let \( G_d \) be the space \( L_2(D_d, \rho_d) \) for a probability density function \( \rho_d \) on \( D_d \). Here we do not need to assume that \( D_d, K_d \) or \( \rho_d \) have the tensor product form. Assume that the operator

\[W_d = S_d^* \circ S_d : F_d \to F_d\]

is compact, where \( S_d \) is the embedding operator into \( G_d \), i.e., \( S_d(f) = f \). Assume also that the ordered eigenvalues \( \{\lambda_n(W_d)\}_{n \geq 1} \) of \( W_d \) satisfy the following condition:

\[\text{there exist } p > 1 \text{ and } B > 0 \text{ such that } \lambda_n(W_d) \leq B \cdot n^{-p} \text{ for all } n.\]

(16)

Then, for every \( n \) and \( \delta \in (0, 1) \), there exists an algorithm \( A_{n,d,\delta} \) with the following three properties:

(P1) \( A_{n,d,\delta} \) uses at most \( n \) function evaluations at points from \( D_d \),
(P2) the range of $A_{n,d,\delta}$ is in $F_d$.
(P3) there exists $c_{\delta,p}$ independent of $n$ and $d$ such that
\[
e^{2}(A_{n,d,\delta};H_d) \leq B \cdot c_{\delta,p} \cdot (n + 1)^{-(1-\delta)p^2/(p+1)}.
\]

We will use this result in the following way. For every non-empty $u \in \Omega(\varepsilon)$, we will replace the algorithms $A_{\omega,\varepsilon}$ in (11) by the algorithms $A_{n,\varepsilon,|u|,\delta}$ for $F_d = H_u$ and $d = |u|$.

First of all note that, due to the assumption $\tau^*(y,\lambda) < 1$, the interval $[\tau^*(y,\lambda), 1)$ is well defined. For a given $u$ and $\tau \in [\tau^*(y,\lambda), 1)$, the eigenvalues $\lambda_{j,u}$ of the operator $W_u$ satisfy
\[
\sum_{j \in \mathbb{N}_+} \lambda_{j,u} = [L(\tau;\lambda)]^{|u|} < \infty.
\]
Let $\{\lambda_n(W_u)\}_{n \geq 1}$ be the ordered sequence of the eigenvalues $\{\lambda_{j,u}\}_{j \in \mathbb{N}^{|u|}}$. Obviously,
\[
\sum_{j=1}^{\infty} |\lambda_n(W_u)|^\tau = \sum_{j \in \mathbb{N}^{|u|}} \lambda_{j,u}^\tau
\]
and therefore
\[
n \cdot [\lambda_n(W_u)]^\tau \leq [L(\tau;\lambda)]^{|u|}, \quad \text{i.e.,} \quad \lambda_n(W_u) \leq [L(\tau;\lambda)]^{|u|/\tau} \cdot n^{-1/\tau}.
\]
This means that the approximation problem over $H_u$ satisfies the assumption (16) with $B = B_u = [L(\tau;\lambda)]^{|u|/\tau}$ and $p = 1/\tau > 1$.

The algorithm $A_{\omega,\varepsilon,|u|,\delta}$ requires to sample the function $f_u$. Since we can only sample the function $f$, we need to apply the formula from [18] which states that
\[
f_u(x, u) = \sum_{v \subseteq u} (-1)^{|u| - |v|} f(x, v).
\]
This means that we can evaluate $f_u$ at a point by evaluating $f$ at $2^{|u|}$ points, each of them with at most $|u|$ active variables.

If $\emptyset \in \Omega(\varepsilon)$ then we set $n_\emptyset = 1$ and $A_{n_\emptyset,\emptyset,\delta}(f) = f(a)$. Finally, consider
\[
A_{\varepsilon,\delta}(f) := \sum_{u \in \Omega(\varepsilon)} A_{n_\varepsilon,\varepsilon,|u|,\delta}(f_u)
\]
with
\[
n_u = n_u(\varepsilon, \delta, \tau) := \left[ (\varepsilon^{-2} \cdot c_{\delta,1/\tau} \cdot \gamma_u \cdot [L(\tau;\lambda)]^{|u|/\tau})^{(\tau + r/2)/(1-\delta)} \right]
\]
for non-empty $u$.

Using (10), (17) and property (P3), we estimate the error of the $A_{\varepsilon,\delta}$ as follows
\[
\|f - A_{\varepsilon,\delta}(f)\|_g^2 = \sum_{u \in \Omega(\varepsilon)} \|f_u - A_{\varepsilon,\delta}(f_u)\|_g^2 + \sum_{u \in \Omega(\varepsilon)} \|f_u\|_g^2
\]
\[
\leq \sum_{u \in \Omega(\varepsilon)} \frac{B_u \cdot c_{\delta,1/\tau}}{(n_u + 1)^{(1-\delta)/\tau + r^2)} \cdot \|f_u\|_H_u^2 + \sum_{u \in \Omega(\varepsilon)} \|f_u\|_H_u^2 \cdot \lambda_1^{|u|}
\]
\[
\leq \sum_{u \in \Omega(\varepsilon)} \varepsilon^2 \cdot \gamma_u^{-1} \cdot \|f_u\|_H_u^2 + \sum_{u \in \Omega(\varepsilon)} \varepsilon^2 \cdot \gamma_u^{-1} \cdot \|f_u\|_H_u^2
\]
\[
= \varepsilon^2 \cdot \|f\|_F^2.
\]
This means that the error of $A_{\varepsilon,\delta}$ does not exceed $\varepsilon$. 
We now analyze the cost of the algorithm $A_{\tau^{\ast}}^{\mathrm{std}}$. As already mentioned, the cost of one sampling of $f_u$ is bounded from above by $2^{\left|u\right|} \cdot \left|\left|u\right|\right|$. Therefore
\[
\text{cost}(A_{\tau^{\ast}}^{\mathrm{std}}) \leq \sum_{u \in \mathcal{U}(\epsilon)} |u| \cdot 2^{|u|} \cdot n_u \leq S(d(\epsilon)) \cdot 2^{d(\epsilon)} \cdot \sum_{u \in \mathcal{U}(\epsilon)} n_u.
\]

Note that
\[
\sum_{u \in \mathcal{U}(\epsilon)} n_u \leq \left(\frac{C_{\tau^{\ast}}}{\epsilon^2}\right)^{(r+\tau^2)/(1-\delta)} \sup_{u \in \mathcal{U}(\epsilon)} \left(\gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}\right)^{(r+\tau^2)/(1-\delta)} \sum_{u \in \mathcal{U}(\epsilon)} \gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}
\]
\[
= \left(\frac{C_{\tau^{\ast}}}{\epsilon^2}\right)^{(r+\tau^2)/(1-\delta)} \cdot C(\tau) \cdot \left[\sup_{u \in \mathcal{U}(\epsilon)} \gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}\right]^{(r+\tau^2)/(1-\delta)}.
\]

From this analysis we easily obtain the following theorem.

**Theorem 3.** Let $\tau^\ast(\gamma, \lambda) < 1$. For any $\tau \in (\tau^\ast(\gamma, \lambda), 1)$ and $\delta \in (0, 1)$, the algorithm given by (18) and (19) satisfies
\[
e(\epsilon, \tau^{\ast}; \mathcal{F}) \leq \epsilon
\]
and its cost is bounded by
\[
\text{cost}(A_{\tau^{\ast}}^{\mathrm{std}}) \leq S(d(\epsilon)) \cdot 2^{d(\epsilon)} \cdot \left(\frac{C_{\tau^{\ast}}}{\epsilon^2}\right)^{(r+\tau^2)/(1-\delta)} \cdot C(\tau) \cdot \left[\sup_{u \in \mathcal{U}(\epsilon)} \gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}\right]^{(r+\tau^2)/(1-\delta)}
\]
\[
\leq S(d(\epsilon)) \cdot 2^{d(\epsilon)} \cdot \left(\frac{C_{\tau^{\ast}}}{\epsilon^2}\right)^{(r+\tau^2)/(1-\delta)} \cdot C(\tau) \cdot \left[\gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}\right]^{(r+\tau^2)/(1-\delta)}
\]
\[
= \Theta\left(\frac{S(d(\epsilon))}{\epsilon^2} \cdot \left(\frac{C_{\tau^{\ast}}}{\epsilon^2}\right)^{(r+\tau^2)/(1-\delta)} \cdot C(\tau) \cdot \left[\gamma_u^{\tau^{\ast}} \cdot \left[L(\tau; \lambda)\right]^{\left|u\right|}\right]^{(r+\tau^2)/(1-\delta)}\right).
\]

where $d(\epsilon)$ is bounded as in (14), i.e., $d(\epsilon) = o(\ln(1/\epsilon))$. In particular,

(i) Let $S(d) = \Theta(e^{k \cdot d})$ for some $k \geq 0$. Then APP is polynomially tractable and, for all $\tau \in (\tau^\ast(\gamma, \lambda), 1)$, we have
\[
\text{comp}(\epsilon; \mathcal{F}) = \Theta(e^{-2(\tau^\ast)^2/(1-\delta)+o(1)}).
\]
The exponent of polynomial tractability is bounded by
\[
p_{\mathrm{std}}^{\tau^\ast(\gamma, \lambda)} \leq 2 \cdot \tau^\ast(\gamma, \lambda)^2 + 2 \cdot [\tau^\ast(\gamma, \lambda)]^2.
\]

(ii) Let $S(d) = \Theta(e^{k \cdot d})$ for some $k \geq 0$ and $c \in (0, 1)$. Then APP is weakly tractable and, for all $\tau \in (\tau^\ast(\gamma, \lambda), 1)$, we have
\[
\text{comp}(\epsilon; \mathcal{F}) = \Theta(e^{-2(\tau^\ast)^2/(1-\delta)+o(1)} \cdot e^{-\beta(\epsilon)}),
\]
where
\[
\beta(\epsilon) := k \cdot \frac{[d(\epsilon)]^c}{\ln(1/\epsilon)} = o(1) \quad \text{as} \quad \epsilon \to 0.
\]

The factors in the big $\Theta$ notation are independent of $\epsilon^{-1}$ and dependent on $\tau, \gamma, \lambda$ and $\delta$.

We comment on the bounds for the exponent $p_{\mathrm{std}}$ of polynomial tractability. For $S(d) = \Theta(d^s)$ with $s \geq 0$ or $S(d) = \Theta(e^{k \cdot d})$ with $k \geq 0$, the bound on $p_{\mathrm{std}}$ does not depend on $s$ nor $k$. It is interesting to compare it with the exponent $p_{\mathrm{all}}$ of polynomial tractability for the class $\Lambda_{\mathrm{all}}$. We have $p_{\mathrm{all}} = 2 \cdot \tau^\ast(\gamma, \lambda)$ and therefore
\[
p_{\mathrm{std}} - p_{\mathrm{all}} \leq 2 \cdot [\tau^\ast(\gamma, \lambda)]^2.
\]
3.2. Case 2: $\tau^*(\gamma, \lambda) \geq 1$ and $L(1; \lambda) < \infty$

We now use results from [11]. Let $F_d$, $K_d$, $\rho_d$, and $W_d$ be as in Case 1. Observe that $L(1; \lambda) < \infty$ implies that $W_d$ has a finite trace. Indeed, as before, let $\{\lambda_n(W_d)\}_{n \geq 1}$ be the sequence of ordered eigenvalues of $W_d$. Then

$$
\text{trace}(W_d) := \sum_{n=1}^{\infty} \lambda_n(W_d) = \int_{D_d} K_d(x, x) \cdot \rho_d(x) \, dx
$$

$$
= \left[ \int_{D} K_H(x, x) \cdot \rho(x) \, dx \right] \mu = [L(1; \lambda)]^\mu < \infty.
$$

From [11, Thm. 1] we know that, for every $n \geq 1$, there exists an algorithm $A_{n,d}$ that satisfies (P1) and (P2). However, instead of (P3) the following condition now holds:

$$
[e(A_{n,d}; H_d)]^2 \leq \min_{k=0,1,\ldots} \left( \lambda_{k+1}(W_d) + \frac{2 \cdot \text{trace}(W_d) \cdot k}{n+1} \right).
$$

As before, we will apply this result to replace the algorithms $A_{\varepsilon, u}^{opt}$ in (11) by the algorithms $A_{\varepsilon, u}^{std}$ which are the algorithms $A_{n,d}$ applied for $F_d = H_d$ with $d = |u|$. More precisely, for $u \in \mathcal{U}(\varepsilon)$, we have

$$
\text{trace}(W_u) = \int_{D^{[u]}} K_u(x, x) \cdot \rho_u(x) \, dx = [L(1; \lambda)]^{|u|} < \infty.
$$

On the other hand, if we take $\tau \geq 1$ then $L(\tau; \lambda) < \infty$ and, as established before,

$$
\lambda_n(W_u) \leq [L(\tau; \lambda)]^{|u|/\tau} \cdot n^{-1/\tau},
$$

where $\lambda_n(W_u)$ stands for the $n$th largest eigenvalue of $W_u$.

We replace the minimum in (22) with respect to $k$ by using a specific value of $k = k_u$,

$$
k_u = k_u(n_u) = \left( n_u + 1 \right)^{\tau/(1+\tau)} \cdot 2^{-\tau/(1+\tau)} \cdot \left( \frac{L(\tau; \lambda)}{[L(1; \lambda)]^\tau} \right)^{|u|/(1+\tau)}
$$

and $n_u$ that will be derived in a moment. That is, we will use the following bound

$$
[e(A_{n,u}^{std}; H_u)]^2 \leq \frac{[L(\tau; \lambda)]^{|u|/\tau}}{(k_u + 1)^{1/\tau}} \cdot \frac{2}{n_u + 1} \cdot \frac{\lambda_{k_u}(W_u) \cdot [L(1; \lambda)]^{|u|}}{n_u + 1}
$$

$$
\leq \frac{2^{2+\tau}}{(n_u + 1)^{1/(1+\tau)}} \cdot \left( L(1; \lambda) \cdot L(\tau; \lambda) \right)^{|u|/(1+\tau)}.
$$

Hence, to guarantee that these errors do not exceed $\varepsilon^2/\gamma_u$, we choose

$$
n_u := \left[ \frac{2^{2+\tau} \cdot \gamma_u^{1+\tau} \cdot [L(1; \lambda) \cdot L(\tau; \lambda)]^{|u|}}{\varepsilon^{2(1+\tau)}} \right].
$$

Then the algorithm

$$
A_{\varepsilon}^{std}(f) := \sum_{u \in \mathcal{U}(\varepsilon)} A_{n_u,u}^{std}(f_u)
$$

has error bounded by $\varepsilon$.

---

3 Theorem 1 in [11] holds for $n \geq 1$ with the factor $1/n$ which we replace here by $2/(n+1)$ since then (22) holds also for $n = 0$. 
As before, we estimate the cost by
\[
\text{cost}(A_{\varepsilon}^{\text{std}}) \leq \frac{2^{2+\tau} \cdot S(d(\varepsilon)) \cdot 2^{d(\varepsilon)}}{\varepsilon^{2}(1+\tau)} \cdot \sum_{u \in \mathbb{U}(\varepsilon)} \gamma_u^{1+\tau} \cdot [L(1; \lambda) \cdot L(\tau; \lambda)]^{[u]} \leq \frac{2^{2+\tau} \cdot S(d(\varepsilon)) \cdot 2^{d(\varepsilon)}}{\varepsilon^{2}(1+\tau)} \cdot C(\tau) \cdot \max_{u \in \mathbb{U}(\varepsilon)} \gamma_u \cdot [L(1; \lambda)]^{[u]}.
\] (25)

To guarantee that \( C(\tau) < \infty \), we take \( \tau > \tau^*(\gamma, \lambda) \). We now show that
\[
\max_{u \in \mathbb{U}(\varepsilon)} \gamma_u \cdot [L(1; \lambda)]^{[u]} = O(\exp(o(\varepsilon^{-1}))) = O((1/\varepsilon)^{o(1)}).
\]

Indeed, we have
\[
\gamma_u \cdot [L(1; \lambda)]^{[u]} \leq \gamma_u \cdot [L(\tau; \lambda)]^{[u]/\tau} \cdot \left[ \frac{L(1; \lambda)}{L(\tau; \lambda)^{1/\tau}} \right]^{[u]}.\]

From Jensen’s inequality we have \( L(1; \lambda)/L(\tau; \lambda)^{1/\tau} \geq 1 \). Furthermore,
\[
\gamma_u \cdot [L(\tau; \lambda)]^{[u]/\tau} \leq C(\tau)^{1/\tau} < \infty.
\]

Therefore the maximum above is bounded by
\[
C(\tau)^{1/\tau} \cdot \left[ \frac{L(1; \lambda)}{L(\tau; \lambda)^{1/\tau}} \right]^{d(\varepsilon)}.
\]

Since \( d(\varepsilon) = o(\ln(1/\varepsilon)) \), the last expression is of the form \( O(\exp(o(\varepsilon^{-1}))) = O((1/\varepsilon)^{o(1)}) \), as claimed. This analysis yields the following theorem.

**Theorem 4.** Let \( \tau^*(\gamma, \lambda) \geq 1 \) and \( L(1; \lambda) < \infty \). For any \( \tau > \tau^*(\gamma, \lambda) \), the algorithm given by (23) and (24) satisfies
\[
e(\mathcal{A}_{\varepsilon}^{\text{std}}; \mathcal{F}) \leq \varepsilon
\]

and
\[
\text{cost}(A_{\varepsilon}^{\text{std}}) \leq \frac{2^{2+\tau} \cdot S(d(\varepsilon)) \cdot 2^{d(\varepsilon)}}{\varepsilon^{2}(1+\tau)} \cdot C(\tau) \cdot \max_{u \in \mathbb{U}(\varepsilon)} \gamma_u \cdot [L(1; \lambda)]^{[u]} = O\left( \frac{S(d(\varepsilon))}{\varepsilon^{2+2-\tau+o(1)}} \right),
\] (26)

where \( d(\varepsilon) = o(\ln(1/\varepsilon)) \), see (14). In particular,

(i) Let \( S(d) = O(e^{k\cdot \varepsilon}) \) for some \( k \geq 0 \). Then APP is polynomially tractable and, for all \( \tau > \tau^*(\gamma, \lambda) \), we have
\[
\text{comp}(\varepsilon; \mathcal{F}) = O\left( e^{-2-2-\tau+o(1)} \right).
\]
The exponent of polynomial tractability is bounded by
\[
p_{\text{std}} \leq 2 + 2 \cdot \tau^*(\gamma, \lambda).
\]

(ii) Let \( S(d) = O\left( e^{k \cdot \varepsilon} \right) \) for some \( k \geq 0 \) and \( c \in (0, 1) \). Then APP is weakly tractable and for all \( \tau > \tau^*(\gamma, \lambda) \) we have
\[
\text{comp}(\varepsilon; \mathcal{F}) = O\left( e^{-2-2-\tau+o(1)} \cdot e^{\beta(\varepsilon)} \right),
\]
where \( \beta(\varepsilon) \) is given by (21).

The factors in the big \( O \) notation above are independent of \( \varepsilon^{-1} \) and dependent on \( \tau, \gamma, \lambda. \)

We now combine Theorems 3 and 4. We see that basically \( \tau^2 \) in Theorem 3 is replaced by 1 in Theorem 4. In what follows it will be convenient to denote the algorithm \( A_{\varepsilon}^{\text{std}} \) as \( A_{\varepsilon}^{\text{std}, 0} \).

---

Wenowcombine Theorems 3 and 4. We see that basically \( \tau^2 \) in Theorem 3 is replaced by 1 in Theorem 4. In what follows it will be convenient to denote the algorithm \( A_{\varepsilon}^{\text{std}} \) as \( A_{\varepsilon}^{\text{std}, 0} \).
Corollary 5. Let $\tau^*(\gamma, \lambda) < \infty$ and $L(1; \lambda) < \infty$. For $\tau^*(\gamma, \lambda) < 1$, let $\tau \in (\tau^*(\gamma, \lambda), 1)$ and $\delta \in (0, 1)$. For $\tau^*(\gamma, \lambda) \geq 1$, let $\tau > \tau^*(\gamma, \lambda)$ and $\delta = 0$. The algorithm $A^\text{std}_{e, \delta}$ has the following properties:

$$e(A^\text{std}_{e, \delta}, \mathcal{F}) \leq \varepsilon,$$

$$\text{cost}(A^\text{std}_{e, \delta}) = \Theta \left( \frac{\varepsilon}{\delta} \right).$$

Consider the cost function $S(d) = O(e^{kd})$.

(i) Then APP is polynomially tractable for the class $\Lambda^{\text{std}}$, and the exponent $p^{\text{std}}$ of polynomial tractability is bounded by

$$p^{\text{std}} \leq p^* := 2 \cdot (\tau^*(\gamma, \lambda) + \min(1, [\tau^*(\gamma, \lambda)]^2)).$$

(ii) The condition $\tau^*(\gamma, \lambda) < \infty$ is necessary for polynomial tractability for all RKHS $H$, whereas the condition $L(1; \lambda) < \infty$ is necessary for some RKHS $H$.

As we shall see, the results presented so far can be strengthened for product and finite-order weights and this is the subject of the next three subsections.

3.3. Product weights with polynomial decay

Product weights are of the form

$$\gamma_u = \prod_{j \in u} \gamma_j$$

for positive numbers $\gamma_j$. Recall that for product weights, we have

$$C(\tau) \leq e^{L(\tau; \lambda) - L(\tau; \gamma)} < \infty \quad \text{when} \quad \tau > \tau^*(\gamma, \lambda) = \max \left( \frac{1}{\text{decay}_\lambda}, \frac{1}{\text{decay}_\gamma} \right).$$

Assume a polynomial decay of the product weights, i.e.,

$$\gamma_j \leq C_\gamma \cdot j^{-\alpha} \quad \text{for all} \quad j \in \mathbb{N}_+$$

for some positive $\alpha$ and $C_\gamma$. Note that $L(1; \lambda) < \infty$ implies that $\lambda_j \leq L(1; \lambda)/j$. Hence, $\{\lambda_j\}$ also decays polynomially. From Case 1 of Section 3.2.1 in [8], we know that there exists a positive $c$, dependent on $C_\gamma$, $\alpha$ and $L(1; \lambda)$, such that

$$d(\varepsilon) \leq C \cdot \frac{\ln(1/\varepsilon)}{\ln(\ln(1/\varepsilon))} \quad \text{for all} \quad \varepsilon \in (0, 1/e). \quad (27)$$

Note that (27) is a better estimate of $d(\varepsilon)$ than (14).

For $\tau^*(\gamma, \lambda) < 1$, we obtain from (20) of Theorem 3

$$\text{cost}(A^\text{std}_{e, \delta}) = \Theta \left( \frac{S(c \cdot \ln(1/\varepsilon)/\ln(\ln(1/\varepsilon)))}{e^{2(1+\alpha)(1-\delta)/\alpha(1)}} \right).$$

For $\tau^*(\gamma, \lambda) \geq 1$ and $L(1; \gamma) < \infty$, we obtain from (26) of Theorem 4

$$\text{cost}(A^\text{std}_{e, \delta}) = \Theta \left( \frac{S(c \cdot \ln(1/\varepsilon)/\ln(\ln(1/\varepsilon)))}{e^{2\tau+2+o(1)}} \right).$$

A more significant difference is when weak tractability is concerned. Indeed, we now have weak tractability when $S(d) = \Theta(e^{k \cdot d^e})$ even for $c = 1$. 
3.4. Product weights with exponential decay

Consider now product weights with an exponential decay,
\[ \gamma_j \leq C_r \cdot r^j \quad \text{for all } j \in \mathbb{N}_+ \]
for \( C_r > 0 \) and \( 0 < r < 1 \). Again, from Case 3 of Section 3.2.3 in [8], we easily conclude that
\[ d(\varepsilon) = 2 \cdot \frac{\ln(1/\varepsilon)}{\ln(1/r)} \cdot (1 + o(1)) \quad \text{as } \varepsilon \to 0. \]
This and Corollary 5 yield
\[ \text{cost}(A_{\varepsilon, \lambda}^{\text{std}}) = O \left( \varepsilon^{2 + \min(1, r^2) + o(\varepsilon) + o(1)} \right). \]

Thus, if \( \$d = O \left( e^{k \cdot d^2} \right) \) for some positive \( k \) then we have polynomial tractability with the tractability exponent bounded by
\[ p_{\text{std}} \leq 2 \cdot \left( \tau^*(\gamma, \lambda) + \min(1, [\tau^*(\gamma, \lambda)]^2) + \frac{2 \cdot k}{\ln(1/r)} \right). \]
If \( \$d = O \left( e^{k \cdot d^2} \right) \) for some \( k > 0 \) and \( c < 2 \) then we have polynomial tractability with the exponent
\[ p_{\text{std}} \leq 2 \cdot \left( \tau^*(\gamma, \lambda) + \min(1, [\tau^*(\gamma, \lambda)]^2) \right). \]

Hence, in the latter case, we obtain the same exponent as in [10, Thm. 5] even though we deal with functions of infinitely many variables and with the function evaluation cost \( \$ \) that may depend more than exponentially on the number of active variables.

Note that even for \( \$d = O \left( e^{k \cdot d^2} \right) \) with \( k < \frac{1}{4 \ln(1/r)} \) we obtain weak tractability.

3.5. Finite-order weights

Finite-order weights are of the form
\[ \gamma_u = 0 \quad \text{for all } |u| > \omega \]
for some \( \omega \geq 1 \). Then obviously \( d(\varepsilon) \leq \omega \) and \( \$d(\varepsilon) \in [1, \$\omega) \). This means that the cost function \$ is not essential and may effect only the multiplicative constant of the complexity.

For finite-order weights we define \( \text{decay}_y \) as follows. We first order the weights such that \( \{\gamma_u\} = \{\gamma_{u_j}\} \) with \( \gamma_{u_j} \geq \gamma_{u_{j+1}} \) and define \( \text{decay}_y \) as the decay of \( \{\gamma_u\} = \{\gamma_{u_j}\} \). It is shown in [8] that, for finite-order weights with \( \sup_u \gamma_u < \infty \), polynomial tractability holds for the class \( A_{\text{all}}^\text{all} \) iff \( \text{decay}_y > 0 \) and \( \text{decay}_\lambda > 0 \). If this holds then \( \tau^*(\gamma, \lambda) = \max(1/\text{decay}_y, 1/\text{decay}_\lambda) \) and
\[ p_{\text{all}} = 2 \cdot \max(1/\text{decay}_y, 1/\text{decay}_\lambda). \]

This condition is also necessary and sufficient for polynomial tractability for \( A_{\text{std}}^\text{std} \) when \( L(1; \lambda) < \infty \); however, the tractability exponent \( p_{\text{std}}^{\text{std}} \) might be larger than \( p_{\text{all}}^{\text{all}} \). More precisely, let \( L(1; \lambda) < \infty \). This implies that \( \text{decay}_\lambda \geq 1 \). Then the problem is polynomially tractable for \( A_{\text{std}}^\text{std} \) iff \( \text{decay}_y > 0 \). When this holds, then
\[ p_{\text{all}}^{\text{all}} \leq p_{\text{std}}^{\text{std}} \leq p_{\text{all}}^{\text{all}} + \left[ \min(1, p_{\text{all}}^{\text{all}}/2) \right]^2. \]

4. Constructive results

The results of this section are based on the Smolyak algorithm for approximating linear tensor product problems, see [12], and the analysis of this algorithm done in [13].
Consider first the problem of approximating the scalar functions \( f \in H \) with the error measured in the \( \rho \)-weighed \( L_2 \) norm. Following [13, Thm. 1], we assume that we know a family of algorithms \( A_\varepsilon \) using standard information such that the following conditions hold for some numbers \( C \cdot q, E, F_0, F_1 \),

\[
\begin{align*}
(\text{A1}) \quad &\| S - A_\varepsilon \| \leq C \cdot q^i \text{ for all } i \geq 0, \text{ where } C \geq 0 \text{ and } q < 1, \\
(\text{A2}) \quad &\| A_\varepsilon - A_{\varepsilon-1} \| \leq E \cdot q^i \text{ for all } i \geq 1, \text{ where } E \geq 0, \\
(\text{A3}) \quad &A_\varepsilon \text{ uses at most } F_0 \cdot (F_1^i - 1) \text{ function values, where } F_0 > 0 \text{ and } F_1 > 1.
\end{align*}
\]

Then, for any \( u \), Theorem 1 in [13] provides a family of algorithms \( A_{\varepsilon,u} \in (0,1) \) for approximating \( f_u \) in the \( \rho_u \)-weighed \( L_2 \) norm such that

\[
\begin{align*}
(\text{T1}) \quad &e(A_{\varepsilon,u}; H_u) \leq \varepsilon, \\
(\text{T2}) \quad &\text{there are positive numbers } \alpha_0, \alpha_1, \alpha_2 \text{ independent of } u \text{ and } \varepsilon \text{ for which the number } n(A_{\varepsilon,u}) \text{ of function values used by the algorithm } A_{\varepsilon,u} \text{ is bounded by }
\end{align*}
\]

\[
\begin{align*}
n(A_{\varepsilon,u}) &\leq \alpha_0 \left( \alpha_1 + \alpha_2 \cdot \frac{\ln(1/\varepsilon)}{|u|} \right) \left( \frac{1}{\varepsilon} \right)^{\kappa} \text{ with } \kappa = \frac{\ln(F_1)}{\ln(1/q)}. \tag{28}
\end{align*}
\]

For \( |u| = 1 \), we use the convention that \( \infty^0 = 1 \).

Clearly, the smaller \( \kappa \) the less expensive the algorithms \( A_{\varepsilon,u} \). This is why we now comment on possible values of \( \kappa \) in relation to the decay of the eigenvalues \( \lambda_j \).

**Remark 6.** Suppose that the eigenvalues \( \lambda_j \) of the operator \( W \) have decay \( \lambda_j = p \). Since we always assume that \( L(1; \gamma) = \sum_{j=1}^{\infty} \lambda_j < \infty \), we have \( p \geq 1 \). Then from [10] for \( p > 1 \) and [11] for \( p = 1 \), we know that, for any positive \( \delta \), the assumptions (A1)–(A3) hold with

\[
\kappa = \begin{cases} 
2 \cdot (p + 1) \cdot (1 + \delta) & \text{if } p > 1, \\
4 & \text{if } p = 1.
\end{cases}
\]

However, for all known reproducing kernels \( K_H \), these assumptions hold with

\[
\kappa = \frac{2}{p}.
\]

Since standard information is not more powerful than linear information, \( 2/p \) is the minimal value for \( \kappa \) and then, standard information is as powerful as linear information. For instance, this holds for the classical Wiener kernel \( K_H(x, y) = \min(x, y) \) mentioned in Example 1 with \( D = [0, 1] \) and \( \rho \equiv 1 \). Then

\[
\lambda_j = \Theta(j^{-2}) \quad \text{and} \quad \kappa = 1.
\]

For \( r \)-folded Wiener kernel

\[
K_H(x, y) = \int_0^{\min(x, y)} \frac{(x - t)^r \cdot (y - t)^r}{|r!|^2} \, dt,
\]

we have

\[
\lambda_j = \Theta(j^{-2(r+1)}) \quad \text{and} \quad \kappa = \frac{1}{r+1}.
\]

It is a major open problem whether \( \kappa = 2/p \) can be taken for all Hilbert spaces with \( L(1; \gamma) < \infty \).

Similarly as in the previous section, we replace the algorithms \( A_{\varepsilon,u}^{\text{opt}} \) by the algorithms \( A_{\varepsilon,\sqrt{\rho_u u}} \), i.e., we define

\[
A_{\varepsilon}^{\text{std}}(f) := f(a) + \sum_{\emptyset \neq u \in \ell(\varepsilon)} A_{\varepsilon,\sqrt{\rho_u u}}(f_u). \tag{29}
\]
Clearly, 

\[ e(A^\text{std}_\varepsilon; F) \leq \varepsilon \]

and the cost of the algorithm \( A^\text{std}_\varepsilon \) is bounded by

\[
\text{cost}(A^\text{std}_\varepsilon) \leq S(0) + \alpha_0 \cdot \sum_{\emptyset \neq u \in U(\varepsilon)} S(|u|) \cdot 2^{|u|} \cdot \left( \alpha_1 + \alpha_2 \cdot \frac{\ln(\sqrt{\gamma_u}/\varepsilon)}{|u| - 1} \right)^{(\kappa + 1) \cdot (|u| - 1)} \cdot \left( \frac{\sqrt{\gamma_u}}{\varepsilon} \right)^\kappa.
\]

Let

\[ \gamma_\infty := \sup_u \gamma_u < \infty. \]

Then we have,

\[
\text{cost}(A^\text{std}_\varepsilon) \leq S(0) + \frac{S(d(\varepsilon))}{e^\kappa} \cdot \alpha_0 \cdot \gamma_\infty^{\kappa/2} \cdot 2^{d(\varepsilon)} \cdot f(\varepsilon) \cdot \sum_{u \in U(\varepsilon)} \gamma_u^{\kappa/2}.
\]

Here, for \( d(\varepsilon) \leq 1 \) we have \( f(\varepsilon) = 1 \). For \( d(\varepsilon) > 1 \) we have

\[
f(\varepsilon) = \max_{d = [1, d(\varepsilon)]} \left( \alpha_1 + \alpha_2 \cdot \frac{\ln(\sqrt{\gamma_\infty} + \ln(1/\varepsilon))}{d - 1} \right)^{(\kappa + 1) \cdot (d - 1)}. \]

It is easy to check that for \( \alpha_1 \geq e \), the maximum with respect to \( d \) is attained for \( d = d(\varepsilon) \). Therefore for all \( d(\varepsilon) \) we have

\[
f(\varepsilon) \leq \left( \max(\alpha_1, e) + \alpha_2 \cdot \frac{\ln(\sqrt{\gamma_\infty} + \ln(1/\varepsilon))}{\max(d(\varepsilon), 1) - 1} \right)^{(\kappa + 1) \cdot (\max(d(\varepsilon), 1) - 1)}.
\]

Since \( x = (1/\varepsilon)^{\ln(x)/\ln(1/\varepsilon)} \) for a positive \( x \), we can rewrite the last bound for \( d(\varepsilon) > 1 \) as

\[
f(\varepsilon) = O((1/\varepsilon)^{\Theta(x_\varepsilon \cdot \ln(1 + \Theta(1/x_\varepsilon)))}) \quad \text{with} \quad x_\varepsilon = (d(\varepsilon) - 1) / \ln(1/\varepsilon).
\]

Recall that for \( \tau^*(y, \lambda) < \infty \) we have \( d(\varepsilon) = o(\ln(1/\varepsilon)) \). In this case \( x_\varepsilon = o(1) \) and therefore

\[
f(\varepsilon) = O((1/\varepsilon)^{\Theta(1)}).
\]

We summarize this in the following theorem.

**Theorem 7.** Let (A1), (A2), and (A3) hold. For \( \kappa \) given by (28) assume that

\[
\sum_{u \in U_p} \gamma_u^{\kappa/2} < \infty,
\]

and let \( \tau^*(y, \lambda) < \infty \). Then the algorithm \( A^\text{std}_\varepsilon \) given by (29) satisfies

\[
e(A^\text{std}_\varepsilon; F) \leq \varepsilon
\]

and

\[
\text{cost}(A^\text{std}_\varepsilon) = O(S(d(\varepsilon)) \cdot e^{-\Theta(1)}) \quad \text{as} \quad \varepsilon \rightarrow 0,
\]

with the factor in the big \( \Theta \) notation independent of \( \varepsilon^{-1} \). In particular,

(i) Let \( S(d) = O(e^{kd}) \) for some \( k \). Then the algorithm \( A^\text{std}_\varepsilon \) has a polynomial cost, and we achieve polynomial tractability with the exponent \( p^\text{std} \leq \kappa \).

(ii) Let \( S(d) = O(e^{kd}) \) for some \( k \). Then the algorithm \( A^\text{std}_\varepsilon \) has a non-exponential cost, and we achieve weak tractability.

Suppose that the algorithms \( A_i \) from (A1)–(A3) are almost optimal for standard information in the scalar case, and that standard information is as powerful as linear information. Then the corresponding parameter \( \kappa \) satisfies \( \comp(\varepsilon; H) = \Theta(e^{-\kappa}) \) and, hence, \( p^\text{all} = p^\text{std} = \kappa \). We summarize this in the following corollary.
Corollary 8. Let $\kappa$ be given by (28). If

$$\kappa = \frac{2}{\text{decay}_\lambda} \quad \text{and} \quad \sum_{u \in U} \gamma_u^{\kappa/2} < \infty$$

then Theorem 7 holds with the smallest possible exponent

$$p^{\text{std}} = p^{\text{all}} = \kappa.$$ 

4.1. Product weights

We now specialize Theorem 7 for product weights assuming that $s(d) = \Theta(e^{kd})$. As in Section 3.3, we first consider product weights with polynomial decay,

$$\gamma_u = \prod_{j \in u} \gamma_j \quad \text{with} \quad \gamma_j \leq C \cdot j^{-\alpha} \quad \text{for all} \quad j \in \mathbb{N}_+$$

for positive $C$ and $\alpha$. This means that $\text{decay}_\gamma \geq 1/\alpha$. We also know that, for product weights with a polynomial decay, we have

$$\tau^*(\gamma, \lambda) = \max \left( \frac{1}{\text{decay}_\lambda}, \frac{1}{\text{decay}_\gamma} \right)$$

and

$$\kappa \geq \frac{2}{\text{decay}_\lambda}.$$

Then

$$\sum_{u \in U(\varepsilon)} \gamma_u^{\kappa/2} \leq \sum_{p=0}^{d(\varepsilon)} \frac{1}{p!} \cdot \left[ \sum_{j=1}^{\infty} \gamma_j^{\kappa/2} \right]^p = \sum_{p=0}^{d(\varepsilon)} \frac{[L(\kappa/2, \gamma)]^p}{p!} \leq e^{L(\kappa/2, \gamma)}.$$ 

Clearly, for

$$\kappa > \frac{2}{\alpha},$$

both $L(\kappa/2, \gamma)$ as well as $\sum_{u \in U(\varepsilon)} \gamma_u^{\kappa/2}$ are finite.

If we assume that the univariate eigenvalues $\lambda_j = \Theta(j^{-p})$ for some $p \geq 1$, then $\text{decay}_\lambda = p$. As we know we then can take $\kappa$ arbitrarily close to $2(p + 1)/p^2$. However, as mentioned in Remark 6, for many kernels we know that the rates of convergence for both classes $A^{\text{all}}$ and $A^{\text{std}}$ are the same. In this case, we can take $\kappa = 2/p$. Now assuming that $\alpha$ in the estimates of $\gamma_j$’s is sharp, we have $\text{decay}_\gamma = \alpha$, and

$$\tau^*(\gamma, \lambda) = \max(1/\alpha, 1/p) \quad \text{and} \quad p^{\text{all}} = 2 \cdot \max(1/\alpha, 1/p).$$

For the class $A^{\text{std}}$ we have

$$p^{\text{std}} \in \left[ p^{\text{all}}, \max \left( \frac{2}{\alpha}, \frac{2}{p} \cdot (p + 1)/p \right) \right],$$

and if we can take $\kappa = 2/p$ or if $2/\alpha > 2/p \cdot (p + 1)/p$ then

$$p^{\text{std}} = p^{\text{all}}.$$ 

We finally consider product weights with an exponential decay,

$$\gamma_j \leq C \cdot r^j \quad \text{for all} \quad j \in \mathbb{N}_+$$

for $C > 0$ and $r \in (0, 1)$. Clearly, $\text{decay}_\gamma = \infty$ and

$$L(\kappa/2, \gamma) \leq \frac{C^{\kappa/2} \cdot \frac{r^{\kappa/2}}{1 - r^{\kappa/2}}}{\infty} < \infty$$

for any $\kappa$. Hence, $\sum_{u \in U(\varepsilon)} \gamma_u^{\kappa/2} < \infty$ for any $\kappa$. 
Let $\lambda_j = \Theta(j^{-p})$. If we can take $\kappa = 2/p$ then Theorem 7 holds with the smallest possible exponent $p^{\text{std}} = p^{\text{all}} = \frac{2}{p}$.

4.2. Finite-order weights

Similarly as before for product weights, assume for simplicity that $\lambda_j = \Theta(j^{-p})$ with $p > 1$, and $\gamma_u = \Theta(j^{-u})$ with $\alpha > 0$. Then for finite-order weights we have

\[
p^{\text{all}} = 2 \max(1/\alpha, 1/p),
\]

\[
p^{\text{std}} \in [p^{\text{all}}, 2 \max(1/\alpha, 1/p \cdot (p + 1)/p)].
\]

Again if we can take $\kappa = 2/p$ or if $2/\alpha > 2/p \cdot (p + 1)/p$ then $p^{\text{std}} = p^{\text{all}}$.

5. Non-constructive versus constructive results

We provide in this section an example which shows that the assumptions needed for non-constructive results in Section 3 are less restrictive than the assumptions for constructive results in Section 4.

Let the weights $\gamma_u$ have the following nested form

\[
\gamma_u = \begin{cases} 
1 & \text{if } u = \{1, 2, \ldots, |u|\}, \\
0 & \text{otherwise}.
\end{cases}
\]

Consider next a space $H$ and a probability density function $\rho$ such that the eigenvalues $\lambda_j$ of the operator $W$ are equal to $\lambda_j = q^j$ for some $q \in (0, 1)$. Clearly, such $H$ and $\rho$ exist.

We begin with the non-constructive results of Section 3. Note that $\mathrm{decay}_\lambda = \infty$. Moreover, $L(\tau; \lambda) = q^\tau/(1 - q^\tau) < \infty$ for any $\tau > 0$. Hence, $C(\tau) < \infty$ iff $q^\tau < 1 - q^\tau$, i.e., $\tau > \ln(2)/\ln(1/q)$. This means that

\[
\tau^*(\gamma, \lambda) = \frac{\ln(2)}{\ln(1/q)} \quad \text{and} \quad p^{\text{std}} \leq \begin{cases} 
2 \cdot \frac{\ln(2)}{\ln(1/q)} + 2 \cdot \frac{\ln^2(2)}{\ln^2(1/q)} & \text{if } q < 1/2, \\
2 \cdot \frac{\ln(2)}{\ln(1/q)} + 2 & \text{otherwise}.
\end{cases}
\]

On the other hand, the constructive results of Section 4 are not applicable since, for any number $\kappa > 0$,

\[
\sum_{u \in \mathcal{U}} \gamma_u^{\kappa/2} = \sum_{k=1}^{\infty} 1 = \infty.
\]

Acknowledgments

We are grateful for the valuable comments of the two anonymous referees.

References


