COMPLEXITY OF FUNCTION APPROXIMATION WITH ERROR ESTIMATION

FRED J. HICKERNELL

Abstract.

1. General Setting

The problem investigated is that of function recovery. Let $\mathcal{H}$ be some Banach (often Hilbert) space of real-valued functions defined on $\mathcal{X}$. Given any $f \in \mathcal{H}$, one obtains n pieces of information about the function and constructs an approximation $\tilde{f} = A(f) \in \mathcal{G}$, where $\mathcal{G}$ is some, perhaps different, normed linear space. The goal is to make $\|f - \tilde{f}\|_{\mathcal{G}}$ as small as possible.

Here we follow the terminology, but not exactly the notation, in (Traub et al, 1988, Section 3.2). Let $\mathcal{A}(\mathcal{H}, \Lambda)$ be the set of all algorithms, A, that produce approximations $\tilde{f} = A(f)$, for any $f \in \mathcal{H}$ based on information of the class Λ. Specifcially, $A(f)$ is some function of $L_1(f), \ldots, L_n(f)$, where the $L_i \in \Lambda$ are real-valued functions defined on $\mathcal{H}$. Two common choices are all function values, Λ^{std}, and all bounded linear functionals, Λ^{lin}. The algorithm is allowed to be adaptive, i.e., the choice of L_2 may depend on the value of $L_1(f)$, the choice of L_3 may depend on the values $L_1(f)$ and $L_2(f)$, etc. Also, $n =: \text{cost}(A, f)$ may be a function of f, but need not be. The choice of L_i, n, the stopping criterion, etc., are all a part of the definition of A, but f is *not* a part of the definition of A.

The cost of an algorithm for some $\mathcal{F} \subseteq \mathcal{H}$ is defined as

$$\text{cost}(A, \mathcal{F}) = \sup_{f \in \mathcal{F}} \text{cost}(A, f). \tag{1}$$

Given an error tolerance $\varepsilon \geq 0$, let $\mathcal{A}(\varepsilon, \mathcal{F}, \Lambda) \subseteq \mathcal{A}(\mathcal{H}, \Lambda)$ denote the set of all algorithms that guarantee an absolute error no larger than the tolerance, i.e.,

$$\|f - A(f)\|_{\mathcal{G}} \leq \varepsilon \qquad \forall f \in \mathcal{F},\ \forall A \in \mathcal{A}(\varepsilon, \mathcal{F}, \Lambda). \tag{2}$$

The *complexity* of the problem is the cost of the cheapest algorithm that attains the error tolerance, i.e.,

$$\begin{aligned} \text{comp}(\varepsilon, \mathcal{F}, \Lambda) &= \min_{A \in \mathcal{A}(\varepsilon, \mathcal{F}, \Lambda)} \text{cost}(A, \mathcal{F}) \\ &= \min \left\{ \sup_{f \in \mathcal{F}} \text{cost}(A, f) : A \in \mathcal{A}(\mathcal{H}, \Lambda), \quad \|f - A(f)\|_{\mathcal{G}} \leq \varepsilon \ \forall f \in \mathcal{F} \right\}. \end{aligned} \tag{3}$$

The value of $\text{comp}(\cdot, \mathcal{F}, \Lambda)$ clearly depends on the problem definition. If one makes the class of information no larger, $\Lambda' \subseteq \Lambda$, the set of possible functions no smaller, $\mathcal{F} \leq \mathcal{F}'$, and the norm on $\mathcal{G}$ no weaker, $\|g\|_{\mathcal{G}} \leq \|g\|_{\mathcal{G}'}, \forall g \in \mathcal{G}$, then the complexity using these primed quantities will be no smaller, i.e.,

$$\text{comp}(\cdot, \mathcal{F}, \Lambda) \leq \text{comp}(\cdot, \mathcal{F}', \Lambda').$$

The goal in information based complexity theory is to determine the complexity of the problem as a power of the error tolerance and up to a leading constant. Since the problem is linear, this complexity also depends on the size or scale of the functions to be approximated. Define a function

$$\text{size} : \mathcal{H} \to [0, +\infty), \qquad \text{size}(cf) = |c| \, \text{size}(f) \quad \forall f \in \mathcal{H}$$

which is some appropriate measure of size or scale of a function, e.g., a norm or semi-norm. Let $\mathcal{B}_\sigma := \{f \in \mathcal{H} : \text{size}(f) \le \sigma\}$ denote the "ball" of functions of size no larger than σ. A desirable complexity result is upper and lower bounds in terms of power of ε^{-1}, e.g.,

$$C_{\text{lo}} \left(\frac{\sigma}{\varepsilon}\right)^{p_{\text{lo}}} \le \text{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda) \le C_{\text{up}} \left(\frac{\sigma}{\varepsilon}\right)^{p_{\text{up}}}, \quad 0 < \varepsilon < \varepsilon_{\text{up}}. \tag{4a}$$

Here the determination of p_{lo} and p_{up} is the most crucial, but often C_{lo}, C_{up}, and ε_0 can be specified explicitly as well. Clearly, $p_{\text{lo}} \le p_{\text{up}}$. Ideally, it may be possible to choose $p_{\text{lo}} = p_{\text{up}} = p$, and this case is described as

$$\text{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda) \asymp \left(\frac{\sigma}{\varepsilon}\right)^{p}, \quad \varepsilon \downarrow 0.$$

Sometimes the complexity result has a companion result that identifies an explicit algorithm $A_{\varepsilon,\sigma} \in \mathcal{A}(\varepsilon, \mathcal{B}_\sigma, \Lambda)$ that attains the optimal complexity order, i.e.,

$$\|f - A_{\varepsilon,\sigma}(f)\|_{\mathcal{G}} \le \varepsilon \quad \forall f \in \mathcal{B}_\sigma \quad \text{and} \quad \text{cost}(A_{\varepsilon,\sigma}, \mathcal{B}_\sigma) \le C_{\text{up},A_{\varepsilon,\sigma}} \left(\frac{\sigma}{\varepsilon}\right)^{p_{\text{up}}} \quad 0 < \varepsilon < \varepsilon_{\text{up}}. \tag{4b}$$

Other times, (4) is non-constructive. Examples of results of the form (4) are given in and below in ???

Although a complexity result such as (4) is valuable in explaining how fast the work required increases as the error tolerance decreases, its usefulness for solving practical problems is somewhat limited. One must know the appropriate value of σ in advance. One must know a priori that $\text{size}(f) \le \sigma$ before one can pick an algorithm $A_{\varepsilon,\sigma} \in \mathcal{A}(\varepsilon, \mathcal{B}_\sigma, \Lambda)$ that guarantees the required error tolerance as in (5b).

Rather than using an absolute error criterion (2), as has been described above, complexity analysis may be done using an error criterion *normalized* by the size of the function:

$$\frac{\|f - A(f)\|_{\mathcal{G}}}{\text{size}(f)} \le \varepsilon.$$

In this normalized error criterion $0/0$ is taken to be 0 by convention. Moreover, the complexity result (4) now superficially loses the σ-dependence:

$$C_{\text{lo}} \left(\frac{1}{\varepsilon}\right)^{p_{\text{lo}}} \le \text{comp}(\varepsilon, \mathcal{H}, \Lambda) \le C_{\text{up}} \left(\frac{1}{\varepsilon}\right)^{p_{\text{up}}}, \quad 0 < \varepsilon < \varepsilon_{\text{up}}, \tag{5a}$$

and there is found an algorithm A_ε satisfying

$$\frac{\|f - A_\varepsilon(f)\|_{\mathcal{G}}}{\text{size}(f)} \le \varepsilon \quad \forall f \in \mathcal{H} \quad \text{and} \quad \text{cost}(A_\varepsilon, \mathcal{H}) \le C_{\text{up},A_\varepsilon} \left(\frac{\sigma}{\varepsilon}\right)^{p_{\text{up}}} \quad 0 < \varepsilon < \varepsilon_{\text{up}}. \tag{5b}$$

A successful algorithm works for all functions in $\mathcal{H}$, and it does not depend on knowing $\text{size}(f)$ in advance. However, since the error criterion depends on the typically unknown $\text{size}(f)$, one is left still not knowing how accurate the the resulting approximation is.

The purpose of this article is to develop algorithms and a complexity theory that do not require a priori knowledge of the size of a function. This requires a generalization of some of the definitions. The cost of an algorithm, A, for functions in some arbitrary set, $\mathcal{F} \subseteq \mathcal{H}$, is now broken down by function size, so that definition (1) is replaced by

$$\text{cost}(A, \mathcal{F}, \sigma) = \sup_{\substack{f \in \mathcal{F} \\ \text{size}(f) \le \sigma}} \text{cost}(A, f). \tag{6a}$$

A scale-invariant set of functions $\mathcal{N}_\tau \subseteq \mathcal{H}$ is identified, i.e., one with the property that

$$f \in \mathcal{N}_\tau \implies cf \in \mathcal{N}_\tau \qquad \forall c \in \mathbb{R}.$$

An algorithm $A_\varepsilon \in \mathcal{A}(\varepsilon, \mathcal{N}_\tau, \Lambda)$ is constructed that satisfies

$$\|f - A_\varepsilon(f)\|_{\mathcal{G}} \le \varepsilon \quad \forall f \in \mathcal{N}_\tau \quad \text{and} \quad \text{cost}(A_\varepsilon, \mathcal{N}_\tau, \sigma) \le C_{\text{up},\tau,A_\varepsilon} \left(\frac{\sigma}{\varepsilon}\right)^{p_{\text{up}}} \quad 0 < \varepsilon < \varepsilon_{\text{up}}. \tag{6b}$$

Note that the definition of this algorithm *does not depend* on knowing size(f) a priori, although naturally its cost *does* depend on size(f).

The set $\mathcal{N}_\tau$ consists of nice, naive or innocent functions, i.e., those for which the algorithm can reliably estimate how many pieces of information are needed to attain the error tolerance. A function outside this set is one that can trick the algorithm into thinking that it is well approximated at a modest cost, when actually more pieces of information are needed. Specifically, a measure of *guile*, nastiness, or trickiness of a function is defined and $\mathcal{N}_\tau$ is defined as the set of functions with bounded guile, namely,

$$\mathcal{N}_\tau = \{f \in \mathcal{H} : \text{guile}(f) \le \tau\}. \tag{6c}$$

As mentioned above, a function may be multiplied by any constant, thus changing its size, but leaving its guile unchanged.

2. $\mathcal{L}_\infty$ Approximation of Univariate Functions with Bounded First Derivatives Using Function Values

Let $\mathcal{H}$ be the space of continuous functions on $[0, 1]$ with $\mathcal{L}_\infty$ first derivatives. Suppose one wants to approximate functions in $\mathcal{H}$ based on function values. The goal is to make the $\mathcal{L}_\infty$ error small, i.e., to find an algorithm, A for which $\|f - A(f)\|_\infty \le \varepsilon$. A non-adaptive algorithm is described and analyzed first, however, as described in the introduction it is successful only for functions whose size is known. Next an adaptive algorithm is constructed from the non-adaptive one. A measure of guile is defined, and functions with moderate guile are guaranteed to be well-approximated by the non-adpative algorithm.

2.1. Non-Adaptive Algorithms.

As a start, consider the algorithm that produces a piecwise constant approximation using function values at $n + 1$ equally spaced data sites, $\left\{x_i = \frac{i}{n}\right\}_{i=0}^n$:

$$\tilde{A}_n(f) = 1_{\left[0,\frac{1}{2n}\right)}(x)\, f(0) + \sum_{i=1}^{n-1} 1_{\left[\frac{2i-1}{2n},\frac{2i+1}{2n}\right)}(x)\, f(x_i) + 1_{\left[\frac{2n-1}{2n},1\right]}(x)\, f(1). \tag{7}$$

This algorithm assigns as the value $\tilde{A}_n(f)(x)$ the value of $f(x_i)$ corresponding to the data site, x_i, nearest x. The algorithm $\tilde{A}_n$ is used to construct an adaptive algorithm to be described later. Note that the cost of this algorithm is $n+1$, not n. Although this algorithm is slightly sub-optimal for fixed sample size, Define the size of a function as the $\mathcal{L}_\infty$ norm of its first derivative, i.e., $\text{size}(f) = \|f'\|_\infty$. Furthermore, as in the previous section, define the ball of functions $\mathcal{B}_\sigma := \{f \in \mathcal{H} : \|f'\|_\infty \le \sigma\}$.

Theorem 1. *The problem of approximating functions of bounded size has complexity*

$$\text{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda^{\text{std}}) = \left\lceil \frac{\sigma}{2\varepsilon} \right\rceil.$$

The algorithm defined in (7) *is nearly optimal, in the sense that it satisfies the error tolerance with only one additional function evaluation than is optimal:*

$$\sup_{f \in \mathcal{B}_\sigma} \left\| f - \tilde{A}_{\lceil \sigma/(2\varepsilon) \rceil}(f) \right\|_\infty \le \varepsilon, \qquad \text{cost}(\tilde{A}_{\lceil \sigma/(2\varepsilon) \rceil}, \mathcal{B}_\sigma) = \left\lceil \frac{\sigma}{2\varepsilon} \right\rceil + 1.$$

Proof. For any function f lying in the ball $\mathcal{B}_\sigma$, and any $x \in [0,1)$, let x_i be the data site nearest x, and so $|x - x_i| \le 1/(2\lceil \sigma/(2\varepsilon) \rceil) \le \varepsilon/\sigma$ and $\tilde{A}_{\lceil \sigma/(2\varepsilon) \rceil}(f)(x) = f(x_i)$. It then follows that

$$\left| f(x) - \tilde{A}_{\lceil \sigma/(2\varepsilon) \rceil}(f)(x) \right| = |f(x) - f(x_i)| = \left| \int_{x_i}^{x} f'(t) \, \mathrm{d}t \right| \le \|f'\|_\infty |x - x_i| \le \sigma \frac{\varepsilon}{\sigma} = \varepsilon.$$

This implies that $\tilde{A}_{\lceil \sigma/(2\varepsilon) \rceil}$ that attains the error tolerance with $\left\lceil \frac{\sigma}{2\varepsilon} \right\rceil + 1$ function evaluations. A similar argument using an algorithm based on function evaluations at $x_i = (2i-1)/n$, $i = 1, \ldots, n$ can be used to show that $\text{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda^{\text{std}}) \le \left\lceil \frac{\sigma}{2\varepsilon} \right\rceil$.

To prove a lower bound, one considers *any* (possibly adaptive) algorithm A using $n \le \left\lceil \frac{\sigma}{2\varepsilon} \right\rceil - 1$ function evaluations. The proof proceeds by constructing a fooling function, $f^* \in \mathcal{B}_\sigma$ for which $\|f^* - A(f^*)\|_\infty > \varepsilon$. Let A be any such algorithm. Since the algorithm may be adaptive, the choice of the point, ξ_i, where the fooling function, f^*, is evaluated, may depend on $(x_1, f^*(\xi_1)), \ldots, (\xi_{i-1}, f^*(\xi_{i-1}))$, for $i = 2, \ldots, n$. By fiat, the fooling function is constructed to vanish at all data sites, ξ_i. The fooling function is a translated and scaled bump. Define a standard triangular bump on $\mathbb{R}$ as

$$\psi(x) := \max(1 - |x|, 0) = \begin{cases} 0, & |x| > 1, \\ 1 - |x|, & |x| \le 1, \end{cases} \tag{8}$$

and note that $\|\psi\|_\infty = \|\psi'\|_\infty = 1$. Given the algorithm A with data sites $\xi_1, \ldots, \xi_n$, there exists at least one point $\xi_0 \in [0,1)$ with $\min_{1 \le i \le n} |\xi_i - \xi_0| \ge 1/(2n)$. Then define the fooling function by

$$f^* : x \mapsto \frac{\sigma}{2n} \psi(2n(x - \xi_0)).$$

By definition, $\pm f^*(\xi_i) = 0$ for $i = 1, \ldots, n$, and $\|\pm f^{*\prime}\|_\infty = \sigma$, so $\pm f^* \in \mathcal{B}_\sigma$. Moreover, $\pm f^*(\xi_i) = \pm \sigma/(2n)$. Since the two functions $\pm f^*$ provide the same data to algorithm A, it follows that $A(f)(\xi_0) = A(-f)(\xi_0)$. The triangle inequality and the strict upper bound on

n then imply that the approximation error must be greater than the desired tolerance.

$$\begin{aligned}
\sup_{f\in\mathcal{B}} \|f - A(f)\|_\infty &\ge \max\left(\|f^* - A(f^*)\|_\infty, \|-f^* - A(-f^*)\|_\infty\right) \\
&\ge \max\left(|f^*(\xi_0) - A(f)(\xi_0)|, |-f^*(\xi_0) - A(-f^*)(\xi_0)|\right) \\
&\ge \frac{1}{2}\left[|f(\xi_0) - A(f^*)(\xi_0)| + |f^*(\xi_0) + A(-f^*)(\xi_0)|\right] \\
&\ge \frac{1}{2}\left|f^*(\xi_0) - A(f^*)(\xi_0) + f^*(\xi_0) + A(-f^*)(\xi_0)\right| \\
&= |f^*(\xi_0)| = \frac{\sigma}{2n} \ge \frac{\sigma}{2\left(\left\lceil \frac{\sigma}{2\varepsilon}\right\rceil - 1\right)} > \varepsilon.
\end{aligned}$$

Thus, $\text{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda^{\text{std}}) > \left\lceil \frac{\sigma}{2\varepsilon}\right\rceil - 1$, which completes the proof. □

2.2. **Adaptive Algorithms.** The complexity result in Theorem 1 requires knowledg of the size of the function to be approximated, i.e, $\text{size}(f) = \|f'\|_\infty$. This is typically not known in advance, however, it can be approximated from the data. Given the data sites $0, 1/n, \ldots, 1$, and the corresponding function values, one may estimate $\text{size}(f)$ in terms of divided differences:

$$\hat{\sigma}_n(f) = \max_{i=1,\ldots,n} n\left|f(i/n) - f((i-1)/n)\right|. \tag{9}$$

Note that $\hat{\sigma}_n(f)$ never overestimates $\text{size}(f)$. The quality of this estimate depends on how much f' changes over a short distance. This motivates the following definition of guile:

$$\text{guile}(f) := \frac{\|f''\|_\infty}{\|f'\|_\infty} = \frac{\|f''\|_\infty}{\text{size}(f)}, \tag{10}$$

with the convention that constant functions have zero guile. The guile of a function is the size of its second derivative relative to the size of its first derivative. This guile remains unchanged when the function is multiplied by any nonzero constant.

A function with small guile has a size that is well approximated by $\hat{\sigma}_n$. Suppose that ζ is some point satisfying $|f'(\zeta)| = \|f'\|_\infty$ and that $(i-1)/n \le \zeta < i/n$. It then follows that $f(i/n) - f((i-1)/n) = f'(\eta)/n$ for some η between $(i-1)/n$ and i/n. Thus,

$$\begin{aligned}
\|f'\|_\infty = |f'(\zeta)| &= \left|f'(\eta) + \int_\eta^\zeta f''(x)\,\mathrm{d}x\right| \le |f'(\eta)| + \left|\int_\eta^\zeta f''(x)\,\mathrm{d}x\right| \\
&\le \hat{\sigma}_n(f) + |\zeta - \eta|\,\|f''\|_\infty \le \hat{\sigma}_n(f) + \frac{\text{guile}(f)\,\|f'\|_\infty}{n}, \\
\|f'\|_\infty &\le \frac{\hat{\sigma}_n(f)}{1 - \text{guile}(f)/n}.
\end{aligned}$$

Letting $\mathcal{N}_\tau = \{f \in \mathcal{H} : \text{guile}(f) \le \tau\}$ now carves out a subset of $\mathcal{H}$ which includes functions of arbitrarily size, but for which that size can be easily bounded empirically as

$$\text{size}(f) \le \frac{\hat{\sigma}_n(f)}{1 - \tau/n} \qquad \forall f \in \mathcal{N}_\tau. \tag{11}$$

This then suggests the following adaptive algorithm that is guaranteed to approximate functions in $\mathcal{N}_\tau$ within a tolerance of ε.

Algorithm 1. *Given positive numbers ε and τ, let $n_1 = \lceil \tau \rceil + 1$, and $j = 1$.*

Step 1. *Let $n = n_1 2^{j-1}$. Evaluate $\hat{\sigma}_n(f)$.*
Step 2. *If*

$$\frac{\hat{\sigma}_n(f)}{2(n-\tau)} \le \varepsilon,$$

then output $A_\varepsilon(f) = \tilde{A}_n(f)$ as the approximation. Else, let $j = j+1$ and to to Step 1.

Theorem 2. Nice *functions may be well approximated at a reasonable cost by A_ε, the adaptive Algorithm 1. In particular,*

$$\sup_{f \in \mathcal{N}_\tau} \|f - A_\varepsilon(f)\|_\infty \le \varepsilon,$$

$$\left\lceil \frac{\sigma}{2\varepsilon} \right\rceil \le \mathrm{comp}(\varepsilon, \mathcal{N}_\tau, \sigma) \le \mathrm{cost}(A_\varepsilon, \mathcal{N}_\tau, \sigma) \le 2\lceil \tau \rceil + \max\left(2, \frac{\sigma}{\varepsilon}\right) + 1.$$

Proof. For any nice function f lying in $\mathcal{N}_\tau$, its size is bounded above by $\hat{\sigma}_n(f)/(1-\tau/n)$ according to (11). Furthermore, by the proof of Theorem 1 it follows that

$$\left\| f - \tilde{A}_n(f) \right\|_\infty \le \frac{\mathrm{size}(f)}{2n} \le \frac{\hat{\sigma}_n(f)}{2(n-\tau)}.$$

Thus, when the stopping criterion in Step 2 is satisfied, the resulting approximation is within the desired tolerance.

The algorithm terminates for the first positive integer j satisfying the inequality in Step 2. Since $\hat{\sigma}(f)$ never overestimates size(f) for all $f \in \mathcal{H}$, this means that

$$\begin{aligned} n \ge \tau + \frac{\hat{\sigma}_n(f)}{2\varepsilon} &\Longleftarrow (\lceil \tau \rceil + 1)2^{j-1} = n_1 2^{j-1} = n \ge \lceil \tau \rceil + \frac{\mathrm{size}(f)}{2\varepsilon} \\ &\Longleftarrow j \ge 2 + \log_2\left(\frac{\lceil \tau \rceil + \frac{\mathrm{size}(f)}{2\varepsilon}}{\lceil \tau \rceil + 1} \right) \\ &\Longleftarrow n \ge 2\lceil \tau \rceil + \max\left(2, \frac{\mathrm{size}(f)}{\varepsilon}\right). \end{aligned}$$

This establishes an upper bound on the cost of the algorithm.

The proof of the lower bound is similar to that in Theorem 1. Consider *any* (possibly adaptive) algorithm A using $n \le \lceil \frac{\sigma}{2\varepsilon} \rceil - 1$ function evaluations. A fooling function $f^* \in \mathcal{N}_\tau$ is constructed to vanish at all data sites, ξ_i. Unlike Theorem 1, the fooling function must be continuously differentiable. The bump function in Theorem 1 is replaced by

$$\phi(x) := [\max(0, 1-x^2)]^2 = \begin{cases} (1-x^2)^2, & |x| \le 1, \\ 0, & |x| > 1, \end{cases} \tag{12}$$

which has first and second derivatives

$$\phi'(x) = \begin{cases} -4x(1-x^2), & |x| \le 1, \\ 0, & |x| > 1, \end{cases} \qquad \phi''(x) = \begin{cases} -4(1-3x^2), & |x| \le 1, \\ 0, & |x| > 1, \end{cases} \tag{13}$$

and note that $\|\phi\|_\infty = 1$, $\|\phi'\|_\infty = 8/(3\sqrt{3})$, and $\|\phi''\|_\infty = 8$. Given the algorithm A with data sites $\xi_1, \ldots, \xi_n$, there exists at least one point $\xi_0 \in [0,1)$ with $\min_{1 \le i \le n} |\xi_i - \xi_0| \ge 1/(2n)$. Then define the fooling function by

$$f^* : x \mapsto \frac{\sigma}{2n} \psi(2n(x - \xi_0)).$$

By definition, $\pm f^*(\xi_i) = 0$ for $i = 1, \dots, n$, and $\|\pm f^{*\prime}\|_\infty = \sigma$, so $\pm f^* \in \mathcal{B}_\sigma$. Moreover, $\pm f^*(\xi_i) = \pm\sigma/(2n)$. Since the two functions $\pm f^*$ provide the same data to algorithm A, it follows that $A(f)(\xi_0) = A(-f)(\xi_0)$. The triangle inequality and the strict upper bound on n then imply that the approximation error must be greater than the desired tolerance.

$$\begin{aligned}
\sup_{f \in \mathcal{B}} \|f - A(f)\|_\infty &\geq \max\left(\|f^* - A(f^*)\|_\infty, \|-f^* - A(-f^*)\|_\infty\right) \\
&\geq \max\left(|f^*(\xi_0) - A(f)(\xi_0)|, |-f^*(\xi_0) - A(-f^*)(\xi_0)|\right) \\
&\geq \frac{1}{2}\left[|f(\xi_0) - A(f^*)(\xi_0)| + |f^*(\xi_0) + A(-f^*)(\xi_0)|\right] \\
&\geq \frac{1}{2}\left|f^*(\xi_0) - A(f^*)(\xi_0) + f^*(\xi_0) + A(-f^*)(\xi_0)\right| \\
&= |f^*(\xi_0)| = \frac{\sigma}{2n} \geq \frac{\sigma}{2\left(\left\lceil \frac{\sigma}{2\varepsilon} \right\rceil - 1\right)} > \varepsilon.
\end{aligned}$$

Thus, $\operatorname{comp}(\varepsilon, \mathcal{B}_\sigma, \Lambda^{\mathrm{std}}) > \left\lceil \frac{\sigma}{2\varepsilon} \right\rceil - 1$, which completes the proof. □

3. Approximation of Multivariate Functions in a Hilbert Space with Continuous Linear Functional Data

Let $\mathcal{L}_2 = \mathcal{L}_{2,\rho}$ be a separable Hilbert space of real-valued functions of interest defined on $\mathcal{X} \subseteq \mathbb{R}^d$. Let $\rho : \mathcal{X} \to [0, \infty)$ be a non-negative weight used to define the $\mathcal{L}_2$ norm. Let $\{\phi_k\}_{k=1}^\infty$ be an orthonormal basis for this Hilbert space, i.e.,

$$\begin{aligned}
\langle f, g \rangle_2 &:= \int_{\mathcal{X}} \overline{f(\boldsymbol{x})} g(\boldsymbol{x})\, \rho(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \qquad \forall f, g \in \mathcal{L}_2, \\
\langle \phi_k, \phi_\ell \rangle_2 &= \int_{\mathcal{X}} \overline{\phi_k(\boldsymbol{x})} \phi_\ell(\boldsymbol{x})\, \rho(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \delta_{k\ell} \qquad \forall k, l \in \mathbb{N}, \\
\hat{f}(k) &:= \langle \phi_k, f \rangle_2 \qquad \forall k \in \mathbb{N}, \\
f &= \sum_{k=1}^\infty \hat{f}(k) \phi_k, \\
\langle f, g \rangle_2 &= \sum_{k=1}^\infty \overline{\hat{f}(k)} \hat{g}(k).
\end{aligned}$$

Let $\mathcal{H}$ be a Hilbert subspace of $\mathcal{L}_2$ such that $\{\phi_k\}_{k=1}^\infty$ is an orthogonal basis for $\mathcal{H}$. Let $w : \mathbb{N} \to [0, \infty)$ be a non-increasing function. The inner product for $\mathcal{H}$ is defined as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{k=1}^\infty \frac{\overline{\hat{f}(k)} \hat{g}(k)}{w(k)}$$

$$\mathcal{H} = \mathcal{H}_w := \left\{ f = \sum_{k=1}^\infty \hat{f}(k) \phi_k : \|f\|_{\mathcal{H}}^2 = \sum_{k=1}^\infty \frac{|\hat{f}(k)|^2}{w(k)} < \infty \right\}.$$

Note that $\{\sqrt{w(k)}\phi_k\}_{k=1}^\infty$ is an orthonormal basis for $\mathcal{H}_w$.

For now, consider the case where one can obtain any bounded linear functionals. In that case, the best approximation is the Fourier series truncated at the first n terms. That is, one should compute the information

$$L_k(f) = \hat{f}(k) = \langle w(k)\phi_k, f\rangle_{\mathcal{H}} = \langle \phi_k, f\rangle_2, \qquad k = 1, \ldots, n,$$

and then form the approximation as follows:

$$\tilde{f}_{\text{trc},n} = A_{\text{trc},n}(f) = \sum_{k=1}^{n} L_k(f)\phi_k = \sum_{k=1}^{n} \hat{f}(k)\phi_k.$$

The error of this approximation bounded as follows by Hölder's inequality:

$$\left\| f - \tilde{f}_{\text{trc},n} \right\|_2^2 = \left\| \sum_{k=n+1}^{\infty} \hat{f}(k)\phi_k \right\|_2^2 = \sum_{k=n+1}^{\infty} \left| \hat{f}(k) \right|^2 = \sum_{k=n+1}^{\infty} \frac{\left| \hat{f}(k) \right|^2}{w(k)} w(k) \le \|f\|_{\mathcal{H}}^2 \sum_{k=n+1}^{\infty} w(k)$$

$$\sup_{0 \ne f \in \mathcal{H}} \frac{\left\| f - \tilde{f}_{\text{trc},n} \right\|_2}{\|f\|_{\mathcal{H}}} = \sqrt{\sum_{k=n+1}^{\infty} w(k)} =: W(n)$$

Letting

This is approximation is optimal in the following sense:

$$\tilde{f}_{\text{trc},n} = \operatorname*{argmin}_{g \in \mathcal{L}_2} \sup_{f \in} \|f - g\|_2$$

References

Traub JF, Wasilkowski GW, Woźniakowski H (1988) Information-Based Complexity. Academic Press, Boston

Room E1-208, Department of Applied Mathematics, Illinois Institute of Technology, 10 W. 32nd St., Chicago, IL 60616