Capturing Ridge Functions in High Dimensions from Point Queries

Albert Cohen, Ingrid Daubechies, Ronald DeVore,
Gerard Kerkyacharian and Dominique Picard [*]

April 15, 2011

Abstract

Constructing a good approximation to a function of many variables suffers from the "curse of dimensionality". Namely, functions on $\mathbb{R}^N$ with smoothness of order s can in general be captured with accuracy at most $O(n^{-s/N})$ using linear spaces or nonlinear manifolds of dimension n. If N is large and s is not, then n has to be chosen inordinately large for good accuracy. The large value of N often precludes reasonable numerical procedures. On the other hand, there is the common belief that real world problems in high dimensions have as their solution, functions which are more amenable to numerical recovery. This has led to the introduction of models for these functions that do not depend on smoothness alone but also involve some form of variable reduction. In these models it is assumed that, although the function depends on N variables, only a small number of them are significant. Another variant of this principle is that the function lives on a low dimensional manifold. Since the dominant variables (respectively the manifold) are unknown, this leads to new problems of how to organize point queries to capture such functions. The present paper studies where to query the values of a ridge function $f(x) = g(a \cdot x)$ when both $a \in \mathbb{R}^N$ and $g \in C[0,1]$ are unknown. We establish estimates on how well f can be approximated using these point queries under the assumptions that $g \in C^s[0,1]$. We also study the role of sparsity or compressibility of a in such query problems.

1 Introduction

We are interested in approximating functions f defined on a domain $\Omega \subset \mathbb{R}^N$ from their point values when the dimension N is large. Such problems arise when modeling physical processes that depend on many variables, for example in learning theory (see e.g. [23]), in modeling physical and biological systems (see e.g. [12]), and in parametric and stochastic PDEs (see e.g. [4]). We shall assume that $\Omega = [0,1]^N$ throughout this paper.

It is well-known that the general problem of approximating a function f of a large number of variables suffers from the so-called "curse of dimensionality". Namely, if all we know about f is that it has a smoothness of order $s > 0$ then the best approximation order we can receive is $O(n^{-s/N})$ where n is the dimension of the underlying approximation process or the number of computations

[*] This research was supported by the Office of Naval Research Contracts ONR-N00014-08-1-1113, ONR N00014-09-1-0107; the AFOSR Contract FA95500910500; the ARO/DoD Contract W911NF-07-1-0185; the NSF Grants DMS 0915231; the French-German PROCOPE contract 11418YB; the Agence Nationale de la Recherche (ANR) project ECHANGE (ANR-08-EMER-006); the excellence chair of the Fondation "Sciences Mathématiques de Paris" held by Ronald DeVore. This publication is based on work supported by Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST).

used to find the approximation. If N is very large then the smoothness would have to be very high to overcome this curse. This fact has led to the search for other reasonable ways to classify functions in high dimension which hopefully represent real world settings. One such approach is to assume that the target function has a sparse representation in some underlying basis. However, the assumption of sparsity is very close if not the same as a smoothness condition in most settings (see e.g. the characterization of nonlinear approximation orders by Besov smoothness [6]). Another approach, which will be the direction taken in this paper, is to assume that most of the variables have no effect in f (or a rather weak effect). This is a common assumption in manifold learning (see for example [5, 13]) and sliced inverse regression or sensitivity analysis in statistics (see e.g. [21]).

Another ingredient in the problem of approximating a high dimensional function concerns the type of information we have (or can ask) about f. There are many possible settings. Our interest will be in the following problem. We are allowed to only ask for a fixed number n of point values of f and from this information, we must construct our approximation. This can be thought of as a problem in directed learning [15] or optimal recovery [22]. Recent results on problems of this type were given in [8] where the assumptions on f were that it depended on a smaller number of unknown coordinate variables. There are two natural types of point query algorithms: non-adaptive and adaptive. In the former, the set of points where f will be queried are set in advance and do not depend on f. In the latter, each new query point is allowed to depend on the previous query points and the value of f at these points. The algorithms put forward in this paper are adaptive.

Certainly, many other possible models for f can be put forward. A quite general assumption would be that $f(x_1, \dots, x_N) = g(Ax)$ where A is a $m \times N$ matrix with m considerably smaller than N. Our primary interest in this paper will be the case where A is $1 \times N$, i.e. $A = a$ is a vector in $\mathbb{R}^N$. This means that f is a ridge function. In addition to our assumption on the form of A, we shall assume some smoothness condition on the underlying function g. This is a common assumption in statistics, where these models are often called *single index models.* Some algorithms have been provided in this context and minimax bounds investigated see for instance [16],[19],[17] and [18]. The main differences with our approach lie in the fact that the points of observations are supposed to be given in advance and not chosen (as here), and the dimension N of the variable x is supposed to be small in the sense that it does not grow with the number of observations (which would correspond to the case $N < n$ here) and does not interfere in the rates. The closest paper to our setting would be [20], where the function g is supposed to be extremely regular.

To put our results in a precise setting we introduce the following class of functions. Given $s > 0$, we define $\mathcal{R}(s)$ to be the set of all ridge functions $f(x) = g(a \cdot x)$ where $g \in C^s[0,1]$ and $a = (a_1, \dots, a_N) \neq 0$ satisfies $a_i \geq 0$, $i = 1, \dots, N$. If $\lambda = \sum_{i=1}^N a_i$, then $f(x) = g(a \cdot x) = \tilde{g}(a \cdot x/\lambda)$ with $\tilde{g}(t) := g(\lambda t)$ and so we can assume without loss of generality that $\sum_{i=1}^N a_i = 1$. It follows that $a \cdot x \in [0,1]$, whenever $x \in \Omega$. We use the following norm on $C^s[0,1]$. If $k < s \leq k+1$ with $k \in \mathbb{N}$, then we define

$$\|g\|_{C^s} := \|g\|_{C^s[0,1]} := |g^{(k)}|_{\mathrm{Lip}(s-k)} + \sum_{j=0}^{k} \|g^{(j)}\|_{C[0,1]}, \tag{1.1}$$

where for $0 < \beta \leq 1$,

$$|g|_{\mathrm{Lip}(\beta)} := \sup_{x \neq y} \frac{|f(x) - f(y)|}{|x-y|^\beta}.$$

We shall also use the following semi-norm on C^s:

$$|g|_{C^s} := |g|_{C^s[0,1]} := |g^{(k)}|_{\mathrm{Lip}(s-k)}. \tag{1.2}$$

The most important coordinates i in a are those where a_i is large. Thus the complexity of f is in part measured by how many coordinates of a are large. To describe this we take the traditional approach of measuring the compressibility of a. It is well known that the compressibility of a is measured in the form of its membership in ℓ_q spaces (or weak ℓ_q spaces). We recall that a is in weak ℓ_q means that

$$\#\{i : a_i \geq \epsilon\} \leq M\epsilon^{-1/q}, \quad \epsilon > 0, \tag{1.3}$$

and the smallest M for which (1.3) holds is the weak ℓ_q norm $\|a\|_{w\ell_q^N}$ of a. Accordingly, we define $\mathcal{R}(s, q; M_0, M_1)$ to be the collection of all ridge functions $f \in \mathcal{R}(s)$ for which

$$\|g\|_{C^s[0,1]} \leq M_0, \quad \|a\|_{w\ell_q^N} \leq M_1. \tag{1.4}$$

Notice that since the vectors a come from a finite dimensional space $\mathbb{R}^N$, they are in all weak ℓ_q^N. Therefore, it is the size of M_1 that is important in what follows.

We study the following fundamental question: Given the knowledge that f is a ridge function and given a target approximation accuracy $\epsilon > 0$, how many point values would we need of f in order to construct an approximation which achieves this accuracy and where should these point queries be chosen? This can be viewed as a problem of optimal recovery (see [22] or [24]), although we do not believe this model class has been treated previously in the literature. We seek query points and algorithms which have demonstrable performance rates for all of the classes $\mathcal{R}(s, q; M_0, M_1)$. In other words we seek algorithms which are universal over these classes and the algorithms do not require any knowledge of s or q.

In §3. we give an algorithm on where to ask for point values of a ridge function f and then show how to construct a good approximation to f from these point queries. This algorithm does not need to know the values of s or q and so it satisfies universality. We shall show that by asking for $\mathcal{O}(L)$ queries we can find $\hat{f}$ such that whenever $f \in \mathcal{R}(s, q; M_0, M_1)$, with $1 < \bar{s} \leq s \leq S$, then

$$\|f - \hat{f}\|_{C(\Omega)} \leq CM_0\Big(L^{-s} + M_1\epsilon(N, L)^{1/q-1}\Big), \tag{1.5}$$

where

$$\epsilon(N, L) := \begin{cases} \frac{1+\log(N/L)}{L}, & L < N, \\ 0, & L \geq N, \end{cases} \tag{1.6}$$

and where C is a constant depending only on $\bar{s}, S$. The first term on the right in (1.5) corresponds to recovering g and the second to recovering a. The following section (§4) analyzes the stability of our algorithm.

Note that the above results only apply when $s > 1$. We do not know how to remove this restriction. However, we do prove in §5 results for the case $s \leq 1$ under the additional assumption that g is monotone.

We finally devote §6 to some concluding remarks and open questions.

2 Approximation preliminaries

We record in this section some well known results about approximation and compressed sensing which we shall utilize in the following sections. Let us first consider approximating functions in $C[0,1]$. Given integers $S > 1$ and $L \geq 2$, we let $h := 1/L$ and consider the space $\mathcal{S}_h$ of piecewise polynomials of degree $S-1$ with equally spaced knots at the points ih, $i = 1, \ldots, L-1$, and having continuous derivatives of order $S-2$. There is a class of linear operators Q_h which map $C[0,1]$ into $\mathcal{S}_h$ called quasi-interpolants that we shall employ. We refer the reader to Chapter 12 of [7] for a construction of these operators. Given a function $g \in C[0,1]$, the application of Q_h uses only the values of g at the points ih, $i = 0, \ldots, L$. The operator Q_h can be chosen to have the following two properties:

Property Q1: Whenever $g \in C^s[0,1]$, $0 < s \leq S$,

$$\|g - Q_h g\|_{C[0,1]} \leq C|g|_{C^s[0,1]} h^s, \tag{2.1}$$

with C a constant depending only on S.[1]

Property Q2: For any $g \in C[0,1]$, we have

$$\|Q_h g\|_{C[0,1]} \leq C \max_{0 \leq i \leq L} |g(ih)|, \tag{2.2}$$

with C again a constant depending only on S.

Secondly, we consider the approximation of vectors from $\mathbb{R}^N$. For each positive integer k, let Σ_k be the set of those $z \in \mathbb{R}^N$ such that at most k of its coordinates are non-zero. Given any $x \in \mathbb{R}^N$, its error of best approximation in ℓ_p^N from Σ_k is

$$\sigma_k(x)_{\ell_p^N} := \inf_{z \in \Sigma_k} \|x - z\|_{\ell_p^N}. \tag{2.3}$$

For ℓ_p^N norms, the best approximation z to x from Σ_k is gotten by retaining the k biggest entries of x in absolute value (with ties handled in an arbitrary way) and making all other entries zero.

There are simple estimates for $\sigma_k(x)$. Among these, we shall use

$$\sigma_k(x)_{\ell_p^N} \leq k^{1/p-1/q} \|x\|_{\ell_q^N}, \tag{2.4}$$

whenever $q \leq p$. Similarly, we have

$$\sigma_k(x)_{\ell_p^N} \leq C k^{1/p-1/q} \|x\|_{w\ell_q^N}, \tag{2.5}$$

whenever $q < p$, with a constant C depending only on p, q.

We shall also need some well known results on compressed sensing. If $L < N$, we let Φ be an $L \times N$ Bernoulli matrix which satisfies the Restricted Isometry Property [2, 1] of order j for

[1] We use the following conventions for constants. Absolute constants are denote by c_0 (when they appear in bounds that hold for sufficiently small constants) or C_0 when they appear in bounds that hold for sufficiently large constants. The constants are updated each time a new condition is imposed on them. Since there will be a finite number of updates, the final update will determine its value. Constants that are not absolute but depend on parameters will be denoted by C and the parameters will be given. We use the same convention on updating the constants C.

all $j \leq c_0/\epsilon(N, L)$ with $c_0 > 0$ a fixed constant. The entries of Φ are realization of i.i.d. random variables which take values $\pm 1/\sqrt{L}$ with probability $1/2$. We denote by $b_1, \dots, b_L$ the rows of Φ. We also require that Φ satisfies the following mapping property stated in Theorem 4.1 of [9]:

Mapping Property: *There is a fixed constant $c_0 > 0$ such that the following holds. If $L < N$ and $y \in \mathbb{R}^L$ satisfies $\|y\|_{\ell_2^L} \leq c_0\sqrt{\epsilon(N, L)}$ and $\|y\|_{\ell_\infty^L} \leq c_0/\sqrt{L}$, then there is an x from the unit ball of ℓ_1^N such that $\Phi x = y$.*

Note that when Φ is a Bernoulli matrix as described above, these properties are satisfied with extremely high probability on the draw (see [9] and [1]).

In compressed sensing, the matrix Φ is used to extract information. If $x \in \mathbb{R}^N$, the vector $y = \Phi x \in \mathbb{R}^L$ is the information captured by Φ. To decode this information, we shall use the ℓ_1 minimization decoder Δ defined by

$$\Delta(y) := \operatorname*{argmin}_{\Phi u = y} \|u\|_{\ell_1}. \tag{2.6}$$

It is shown in [3] that the encoding-decoding pair (Φ, Δ) has the following instance-optimaliy in ℓ_1^N: For any $x \in \mathbb{R}^N$, we have

$$\|x - \Delta(\Phi x)\|_{\ell_1^N} \leq C_0 \sigma_k(x)_{\ell_1^N}, \tag{2.7}$$

for all $k \leq C_0/\epsilon(N, L)$. Of course, when $L \geq N$, we exactly recover each $x \in \mathbb{R}^N$.

3 An adaptive query algorithm

In this section, we give an algorithm that adaptively queries a ridge function by its point values. The algorithm we give requires us to know that $f \in \mathcal{R}(s)$ with $\bar{s} + 1 \leq s \leq S$ where $\bar{s}, S > 0$ and S is an integer. However, we do not need to know the value of s. It will have three main steps for querying f to extract the information we need. To describe these, we define (as before) $h := 1/L$. Let us notice that asking for the value of f at a point $t(1, \dots, 1)$ gives the value of g at t.

QSTEP1: Evaluate f at base points. We ask for the values of f at the points $\mathcal{B} := \{ih(1, 1, \dots, 1) : \ i = 0, 1, \dots, L\}$ that we refer to as *base points*. This information determines g at the points $t_i := ih$, $i = 0, 1, \dots, L$.

QSTEP 2: Find an interval of large deviation. Let $A := \max_{0 \leq i < j \leq L} \frac{|g(t_i) - g(t_j)|}{|t_i - t_j|}$ and take a pair of points $t_i < t_j$ which assume A. Let $I_0 := [t_i, t_j]$. We ask for the value of f at the point $t(1, \dots, 1)$, $t := (t_i + t_j)/2$, which gives us g at the midpoint t of I_0. If $|g(t_i) - g(t)| \geq |g(t_j) - g(t)|$, we define I_1 as $[t_i, t]$, otherwise we define I_1 as $[t, t_j]$. In either case, the divided difference of g over the two endpoints of I_1 is larger than A. We continue in this way and define $I_2, \dots, I_m$, where

$$m := \lceil \kappa \log_2 L \rceil \ \text{ and } \ \kappa := \frac{2S}{\bar{s}}.$$

Now, the divided difference of g at the two endpoints of $I_m = [\alpha_0, \alpha_1]$ is at least as large as A. So there is a point $\xi_0 \in I_m$ where

$$\frac{|g(\alpha_1) - g(\alpha_0)|}{\alpha_1 - \alpha_0} = |g'(\xi_0)| \geq A. \tag{3.1}$$

We denote by η the midpoint of I_m and by $\delta := |I_m| \le L^{-\kappa}|I_0| \le L^{-\kappa}$ the length of I_m. We finally ask for the value of f at $\eta(1,\dots,1)$ which gives us the value of g at η.

QSTEP3: Query f at padding points. First consider the case $L < N$. The row vectors b_i, $i = 1,\dots,L$, which make up the $L \times N$ Bernoulli matrix satisfy $|b_i \cdot a| \le 1/\sqrt{L}$. We now ask for the value of f at the points $\eta(1,1,\dots,1) + \mu b_i$, $i = 1,\dots,L$, where $\mu := \frac{\sqrt{L}\delta}{2}$. These queries in turn gives the value $g(\eta + \mu b_i \cdot a)$, $i = 1,\dots,L$. All of the points $\eta + \mu b_i \cdot a$ are in I_m because of the definition of μ. In the case $L \ge N$, we ask for the value of f at each point $\alpha_0(1,\dots,1) + \delta e_i$, $i = 1,\dots,N$, where

$$e_i := (0,\cdots,0,1,0,\cdots,0),$$

is the standard Kronecker vector with 1 at position i. This gives the value of g at the points $\alpha_0 + \delta a_i$, $i = 1,\dots,N$. Each of these points is again in I_m.

In summary, we have asked for $L+1$ values of f in **QSTEP1**, $\lceil \kappa \log_2 L \rceil$ point values in **QSTEP2**, and at most L point values in **QSTEP3**; thus a total of at most $2L + 1 + \lceil \kappa \log_2 L \rceil = \mathcal{O}(L)$ point values in all. Next, we describe how we construct an approximation $\hat{f}$ to f from the information we have drawn from f. This is done in two steps.

RSTEP1: Approximating g from the retrieved information. Since $f(ih,\cdots,ih) = g(ih)$ for $i = 0,\cdots,L$, we can construct from the values drawn in **QSTEP 1** the approximation $\hat{g} := Q_h(g)$ to g with Q_h the quasi-interpolant operator. From (2.1) we obtain for each $g \in \mathcal{R}(s,q,M_0,M_1)$,

$$\|g - \hat{g}\|_{C[0,1]} \le CM_0 h^s = CM_0 L^{-s}. \tag{3.2}$$

with C depending only on S.

RSTEP2: Approximating a from the retrieved information. We first consider the case $L < N$ and observe that the information we have drawn in **QSTEP3** allows us to approximate the vector $y := \Phi a \in \mathbb{R}^L$, where $y_i := b_i \cdot a$, $i = 1,\dots,L$. Indeed, for each $i = 1,2,\dots,L$, from the information we have in hand, we can compute

$$\hat{y}_i := \frac{2}{\sqrt{L}}\Big[\frac{g(\eta + \mu b_i \cdot a) - g(\eta)}{g(\alpha_0 + \delta) - g(\alpha_0)}\Big] = \frac{2}{\sqrt{L}}\Big[\frac{g'(\xi_1)\mu b_i \cdot a}{g'(\xi_0)\delta}\Big] = b_i \cdot a\Big[1 + \frac{g'(\xi_1) - g'(\xi_0)}{g'(\xi_0)}\Big], \tag{3.3}$$

because $\mu/\delta = \sqrt{L}/2$. Since $|g'(\xi_0)| \ge A$ and g' is in $\mathrm{Lip}(\bar{s}, M_0)$, we have

$$\Big|\frac{g'(\xi_1) - g'(\xi_0)}{g'(\xi_0)}\Big| \le M_0 A^{-1} L^{-\kappa\bar{s}} =: \gamma. \tag{3.4}$$

This means that we have the following estimate for how well $\hat{y}_i$ approximates the true value y_i:

$$|y_i - \hat{y}_i| \le |y_i|\gamma. \tag{3.5}$$

We apply the ℓ_1-minimization decoder Δ to the vector $\hat{y} := (\hat{y}_i)_{i=1}^L$. This gives $\tilde{a} := \Delta(\hat{y})$.

In the case $L \ge N$, using the information we have drawn in **QSTEP3**, we compute

$$\hat{y}_i := \Big[\frac{g(\alpha_0 + \delta a_i) - g(\alpha_0)}{g(\alpha_0 + \delta) - g(\alpha_0)}\Big] = \Big[\frac{g'(\xi_1) a_i \delta}{g'(\xi_0)\delta}\Big] = a_i[1 + \gamma_i], \tag{3.6}$$

where $|\gamma_i| \le \gamma$, $i = 1, \dots, N$. No decoding is needed in this case and we simply set $\tilde{a}_i = \hat{y}_i = a_i(1 + \gamma_i)$.

The following lemma shows that $\tilde{a}$ is a good approximation to a. We recall the definition of $\epsilon(N, L)$ given in (1.6).

Lemma 3.1 *There is an absolute constant C_0 such that the following holds. If $a \in \ell_q$ with $0 < q < 1$, then*

$$\|a - \tilde{a}\|_{\ell_1^N} \le C_0\Big(\|a\|_{\ell_q^N} \epsilon(N, L)^{1/q-1} + \sqrt{L}\gamma\Big). \tag{3.7}$$

In the case $a \in w\ell_q^N$, we have

$$\|a - \tilde{a}\|_{\ell_1^N} \le C\Big(\|a\|_{w\ell_q^N} \epsilon(N, L)^{1/q-1} + \sqrt{L}\gamma\Big), \tag{3.8}$$

with C now depending on p, q.

Proof: In the case $L \ge N$, we have

$$\|a - \tilde{a}\|_{\ell_1^N} \le \sum_{i=1}^{N} |a_i - \tilde{a}_i| = \sum_{i=1}^{N} |a_i||\gamma_i| \le \gamma, \tag{3.9}$$

and (3.7) and (3.8) hold with no assumptions on a.

In the case $L < N$, we follow ideas from [9]. We know that $|y_i| = |a \cdot b_i| \le L^{-1/2}$, $i = 1, \dots, L$. Therefore, $\|y - \hat{y}\|_{\ell_\infty^L} \le L^{-1/2}\gamma$ and $\|y - \hat{y}\|_{\ell_2^L} \le \gamma$. We can apply the mapping property to find a $z \in \mathbb{R}^N$ such that $\Phi(z) = \hat{y} - y$ and $\|z\|_{\ell_1^N} \le c_0^{-1}\sqrt{L}\gamma$. It follows that $\Phi(a + z) = \hat{y}$ and so $\Delta(\Phi(a + z)) = \tilde{a}$. Using the instance-optimality, we find for any $k \le C_0/\varepsilon(N, L)$

$$\|a + z - \tilde{a}\|_{\ell_1^N} \le C_0 \sigma_k(a + z)_{\ell_1^N} \le C_0(\sigma_k(a) + \|z\|_{\ell_1^N}). \tag{3.10}$$

Since $\|z\|_{\ell_1^N} \le c_0^{-1}\sqrt{L}\gamma$, taking the largest possible value for k and using (2.4)(respectively (2.5)), we arrive at (3.7) (respectively (3.8)). □

The vector $\tilde{a}$ need not have positive coordinates and also need not have ℓ_1 norm one. This defect can be remedied as follows, up to doubling the constants C_0 and C in the estimates (3.7) and (3.8). We first let $a'_i := \max(\tilde{a}_i, 0)$, $i = 1, \dots, N$. Then a' is clearly a better ℓ_1^N approximation to a than $\tilde{a}$, so that we have

$$\|a - a'\|_{\ell_1^N} \le R,$$

with R the right hand side of (3.7) or (3.8). Finally we define $\hat{a} := a'/\|a'\|_{\ell_1^N}$. We can write

$$\|a - \hat{a}\|_{\ell_1^N} \le \|a - a'\|_{\ell_1^N} + \|a' - \frac{a'}{\|a'\|_{\ell_1^N}}\|_{\ell_1^N} \le R + |1 - \|a'\|_{\ell_1^N}| \le 2R.$$

Note that if $a' = 0$ we reach the same conclusion by taking for $\hat{a}$ any vector of positive coordinates and ℓ_1 norm one.

So we have

$$\|a - \hat{a}\|_{\ell_1^N} \le C_0\Big(\|a\|_{\ell_q^N} \epsilon(N, L)^{1/q-1} + \sqrt{L}\gamma\Big) \tag{3.11}$$

and a similar estimate with the weak ℓ_q^N norm on the right.

With these estimates in hand, we can now define our approximation to f. Let us define

$$\hat{f}(x) := \hat{g}(\hat{a} \cdot x). \tag{3.12}$$

Theorem 3.2 *The algorithm described above uses* $2L+1+\lceil \kappa \log_2 L \rceil$ *queries. If* $f \in \mathcal{R}(s, q; M_0, M_1)$, *then the algorithm gives an approximation* $\hat{f}$ *defined in* (3.12) *which satisfies for all* $L \geq 1$,

$$\|f - \hat{f}\|_{C(\Omega)} \leq CM_0 \Big(L^{-s} + C_1 M_1 \epsilon(N, L)^{1/q-1} \Big), \tag{3.13}$$

where C *is a constant depending only on* $\bar{s}$ *and* S *and* C_1 *depends on* q.

Proof: Recalling the definition of A, we distinguish between two cases.

If $A \leq M_0 L^{-s}$, then there is a constant c such that $|g(ih) - c| \leq A/2 \leq M_0 L^{-s}/2$, $i = 0, 1, \ldots, L$. From Properties Q1 and Q2, we find

$$\|\hat{g} - c\|_{C[0,1]} = \|Q_h g - Q_h c\|_{C[0,1]} \leq CM_0 L^{-s}$$

Hence,

$$\|g - c\|_{C[0,1]} \leq \|(g - c) - Q_h(g - c)\|_{C[0,1]} + \|Q_h(g - c)\|_{C[0,1]} \leq CM_0 L^{-s} + CM_0 L^{-s}. \tag{3.14}$$

It follows that

$$\|g(a \cdot x) - \hat{g}(\hat{a} \cdot x)\|_{C(\Omega)} \leq \|g - c\|_{C[0,1]} + \|\hat{g} - c\|_{C[0,1]} \leq CM_0 L^{-s}. \tag{3.15}$$

and so we have proven the theorem in this case.

If $A > M_0 L^{-s}$, then for any $x \in \Omega$, we have from the previous lemma,

$$\begin{aligned} |f(x) - \hat{f}(x)| &\leq |g(a \cdot x) - g(\hat{a} \cdot x)| + |g(\hat{a} \cdot x) - \hat{g}(\hat{a} \cdot x)| \\ &\leq M_0 \|a - \hat{a}\|_{\ell_1^N} + \|g - \hat{g}\|_{C[0,1]} \\ &\leq CM_0 \Big(M_1 \epsilon(N, L)^{1/q-1} + \sqrt{L}\gamma \Big) + CM_0 L^{-s}. \end{aligned}$$

It remains to bound $\sqrt{L}\gamma$. Since $A^{-1} \leq M_0^{-1} L^s$, we have

$$\sqrt{L}\gamma = \sqrt{L} A^{-1} M_0 L^{-\kappa \bar{s}} \leq L^{-\kappa \bar{s} + s} \leq L^{-s}, \tag{3.16}$$

where we have used the definition of κ. Therefore, we have proven the theorem. □

4 Stability of the algorithm

The analysis of the performance of our algorithm given in Theorem 3.2 assumes that, when queried, we receive the exact values of f. This in turn gives that the value of g at the corresponding point is also exact. In this section, we shall show that the conclusion of Theorem 3.2 remains valid even when these values are only given to a given accuracy τ, provided τ is suitably small, in the sense that $\tau \leq CL^{-r}$ for a certain $r \geq s$. This assumption on τ is not the best one can hope for, namely $\tau \leq CL^{-s}$. We leave open the possibility that other algorithms may work even with weaker assumption on the perturbation error τ.

Thus, we assume that in place of the values of g we receive the values of a function $\tilde{g}$ with $\|g-\tilde{g}\|_{C[0,1]} \leq \tau$. As before, we assume that $f \in \mathcal{R}(s)$ with $\bar{s}+1 \leq s \leq S$ where $\bar{s}, S > 0$ and S is an integer. The main steps of the algorithm remain unchanged, except that now we define the value of κ introduced in **QSTEP2** to be slightly larger:

$$\kappa := \frac{2S+1/2}{\bar{s}}. \tag{4.1}$$

Therefore, we still have $\mathcal{O}(L)$ point value queries.

Theorem 4.1 *Suppose that in the execution of our algorithm, we receive the values of f only to accuracy τ. That is, when queried for the value of f at any point x, we receive instead the value $\tilde{f}(x)$ satisfying $|f(x)-\tilde{f}(x)|) \leq \tau$. Then, if $f \in \mathcal{R}(s,q;M_0,M_1)$ and*

$$\tau \leq \frac{M_0}{6} L^{-2S-\kappa-3/2} \tag{4.2}$$

the output $\hat{f}$ of the algorithm satisfies

$$\|f-\hat{f}\|_{C(\Omega)} \leq CM_0\Big(L^{-s} + C_1 M_1 \epsilon(N,L)^{1/q-1}\Big), \tag{4.3}$$

where C is a constant depending only on $\bar{s}$, S and the constant of **Property Q2***, and C_1 depends on q.*

Proof: We first examine the effect on the output $\hat{g}$. Since now we receive the values of $\tilde{g}(t_i)$ in **QSTEP1**, the approximation will be $\hat{g} = Q_h(\tilde{g})$. By **Property Q2**, we have

$$\|Q_h(g) - Q_h(\tilde{g})\|_{C[0,1]} \leq C\|g-\tilde{g}\|_{C[0,1]} \leq C\tau,$$

and therefore (3.2) would be replaced by

$$\|g-\hat{g}\|_{C[0,1]} \leq CM_0(L^{-s}+\tau). \tag{4.4}$$

The effect of imprecise evaluations on the estimation of a is more severe as we now discuss. We continue the proof only in the case $L < N$. Similar arguments hold for $L \geq N$ and are left to the reader. In **QSTEP2**, the algorithm would compute in place of A, the number $\tilde{A} := \max\limits_{0\leq i<j\leq L} \dfrac{|\tilde{g}(t_i)-\tilde{g}(t_j)|}{|t_i-t_j|}$. now based on the received values of $\tilde{g}(t_i)$, $i=0,\dots,L$.

The algorithm would next choose the points $t_{\tilde{i}}, t_{\tilde{j}}$ in place of t_i, t_j and proceed to do the subdivision as called for in **QSTEP2**. The result is to end up with an interval $\tilde{I}_m = [\tilde{\alpha}_0, \tilde{\alpha}_1]$, where as before

$$m := \lceil \kappa \log_2(L) \rceil,$$

but with the update value of κ from (4.1). By construction, the length $\tilde{\delta} := |\tilde{I}_m|$ of this interval satisfies

$$\frac{1}{2}L^{-\kappa-1} \leq \tilde{\delta} \leq L^{-\kappa}. \tag{4.5}$$

We now have

$$|\tilde{g}(\tilde{\alpha}_1) - \tilde{g}(\tilde{\alpha}_0)| \geq \tilde{A}|\tilde{\alpha}_1 - \tilde{\alpha}_0| = \tilde{A}\tilde{\delta}. \tag{4.6}$$

By the mean value theorem, $|g(\tilde{\alpha}_1) - g(\tilde{\alpha}_0)| = |g'(\xi_0)|\tilde{\delta}$ with $\xi_0 \in \tilde{I}_m$. Therefore, from (4.6), we obtain

$$|g'(\xi_0)|\tilde{\delta} \geq \tilde{A}\tilde{\delta} - 2\tau. \tag{4.7}$$

As before, we denote by $\tilde{\eta}$ the center of $\tilde{I}_m$ and $\tilde{\mu} := \sqrt{L}\tilde{\delta}/2$.

For our approximation to $y_i = b_i \cdot a$, the algorithm would compute

$$\tilde{y}_i := \frac{2}{\sqrt{L}}\Big[\frac{\tilde{g}(\tilde{\eta}+\tilde{\mu}y_i) - \tilde{g}(\tilde{\eta})}{\tilde{g}(\tilde{\alpha}_0+\tilde{\delta}) - \tilde{g}(\tilde{\alpha}_0)}\Big] = \frac{2}{\sqrt{L}}\Big[\frac{g(\tilde{\eta}+\tilde{\mu}y_i) - g(\tilde{\eta}) + \beta_1}{g(\tilde{\alpha}_0+\tilde{\delta}) - g(\tilde{\alpha}_0) + \beta_2}\Big] = \frac{2}{\sqrt{L}}\Big[\frac{g'(\xi_1)\tilde{\mu}y_i + \beta_1}{g'(\xi_0)\tilde{\delta} + \beta_2}\Big],$$

where $\xi_1 \in \tilde{I}_m$ and $|\beta_1|, |\beta_2| \leq 2\tau$. The algorithm then decodes to find $\bar{a} := \Delta(\tilde{y})$ and modifies this vector to make its entries nonnegative and sum to one thereby receiving the vector $\tilde{a}$. It then takes $\hat{f}(x) := \tilde{g}(\tilde{a}\cdot x)$ as the output approximation to f.

As in the proof of Theorem 3.2, we consider two cases. The first case is when $\tilde{A} \leq M_0 L^{-s}$. In this case, it does not matter if $\tilde{a}$ approximates a well or not, and we reach (4.3) in a similar way to the proof of Theorem 3.2.

The second case is when $\tilde{A} \geq M_0 L^{-s}$. In this case, we need to see how well $\tilde{a}$ approximates a. Note that from the assumption on τ, we have

$$\tau \leq \tilde{A}\tilde{\delta}/6, \tag{4.8}$$

which combined with (4.7) implies

$$|g'(\xi_0)|\tilde{\delta} \geq 2\tilde{A}\tilde{\delta}/3. \tag{4.9}$$

where the first inequality uses (4.7). We divide the numerator and denominator of the right side by $g'(\xi_0)\tilde{\delta}$ and obtain

$$\tilde{y}_i = \frac{y_i\Big(1 + \frac{g'(\xi_1)-g'(\xi_0)}{g'(\xi_0)}\Big) + \frac{2L^{-1/2}\beta_1}{g'(\xi_0)\tilde{\delta}}}{1 + \frac{\beta_2}{g'(\xi_0)\tilde{\delta}}}. \tag{4.10}$$

Therefore,

$$y_i - \tilde{y}_i = \frac{y_i\Big(\frac{\beta_2}{g'(\xi_0)\tilde{\delta}} - \frac{g'(\xi_1)-g'(\xi_0)}{g'(\xi_0)}\Big) - \frac{2L^{-1/2}\beta_1}{g'(\xi_0)\tilde{\delta}}}{1 + \frac{\beta_2}{g'(\xi_0)\tilde{\delta}}}.$$

By assumption, $g \in \mathrm{Lip}(\bar{s}, M_0)$ and so $|g(\xi_1) - g(\xi_0)| \leq M_0\tilde{\delta}^{\bar{s}}$. Since $|y_i| \leq L^{-1/2}$, this leads to the bound

$$|y_i - \tilde{y}_i| \leq L^{-1/2}\frac{6\tau + M_0\tilde{\delta}^{\bar{s}+1}}{\Big||g'(\xi_0)\tilde{\delta}| - 2\tau\Big|} \leq L^{-1/2}\frac{6\tau + M_0\tilde{\delta}^{\bar{s}+1}}{\tilde{A}\tilde{\delta}/3} \leq 18\tau M_0^{-1}L^{s-1/2}\tilde{\delta}^{-1} + 3L^{s-1/2}\tilde{\delta}^{\bar{s}} \tag{4.11}$$

where we have used (4.7) to bound the denominator. Since $\tilde{\delta} \leq L^{-\kappa} = L^{-\frac{2S+1/2}{\bar{s}}}$, the second term in the right side is bounded by $3L^{-2S+s-1} \leq 3L^{-s-1}$. We use the assumption (4.2) on τ and the bound $\tilde{\delta} \geq (1/2)L^{-\kappa-1}$ to bound the first term on the right side of (4.11) by

$$36\tau M_0^{-1}L^{s+\kappa+1/2} \leq 36L^{-2S+s-1} \leq 36L^{-s-1}.$$

Hence,

$$|y_i - \tilde{y}_i| \leq 39L^{-s-1}, \quad i = 1, \dots, L \tag{4.12}$$

This in turn gives

$$\|y-\tilde{y}\|_{\ell_\infty} \le 39L^{-s-1} \quad \text{and} \quad \|y-\tilde{y}\|_{\ell_2} \le 39L^{-s-1/2}. \tag{4.13}$$

We can now apply the same proof as in Lemma 3.1 to conclude that

$$\|a-\tilde{a}\|_{\ell_1} \le C_0\Big(\|a\|_{\ell_q^N}\epsilon(N,L)^{1/q-1} + L^{-s}\Big). \tag{4.14}$$

Again the vector $\tilde{a}$ can be modified to have nonnegative entries which sum to one while retaining (4.14) (with a change in C_0).

With these two bounds on the approximation of g and a, we are in the same position as in the proof of Theorem 3.2. Therefore, the proof can now be completed exactly as in the proof of Theorem 3.2. □

5 The case $0 < s \le 1$

Up to this point, we have assumed that the function g belongs to $C^s[0,1]$ with $s > 1$. Our previous results do not apply if $s \le 1$. In this section, we shall remedy this, to some extent, by treating the case $g \in C^s$, $0 < s \le 1$. However, we shall make the additional assumption that g is monotone. So, throughout this section, we assume that g is monotone and in $C^s[0,1]$ for some $0 < s \le 1$ which is unknown to us.

Given $L \ge 2$, we define $n := n(L) := \lceil 8L\log_2 L\rceil$. Our first goal is to define a query procedure which finds a partition $\mathcal{I}$ of $[0,1]$ by n intervals I_j, $j = 1,\dots,n$, such that g changes more or less equally on each of these intervals. We do this by adaptive splitting. To begin, we define $\mathcal{I}_1 := \{[0,1/2],[1/2,1]\}$. We recall that asking for the values of f at a point $t(1,\dots,1)$ gives the value of g at the point t, so we can think of such queries as asking for the values of g. We ask for the values of g at $0, 1/2, 1$ and examine the change of g on each of the two intervals in $\mathcal{I}_1$. We choose the interval which has the largest change (with ties broken here and later, by taking the left most interval) and divide it into its two dyadic children. At the general step k of the adaptive algorithm, we consider the current set of $k+1$ intervals and determine the interval on which g has maximal change. We subdivide this interval and ask for the value of g at the new end point. We apply this adaptive subdivision $n-2$ times until we arrive at our final partition $\mathcal{I} := \mathcal{I}(f,h) := \{I_1,\dots,I_n\}$, where the intervals are written from right to left. To create this collection of intervals, we have asked for the value of g at at most $n+1$ points.

The following lemma gives a bound for the change of g on each of the intervals in $\mathcal{I}$.

Lemma 5.1 *If g is monotone and $g \in C^s[0,1]$ for some $0 < s \le 1$, then for any $L \ge 2\max(M,1)/s$ and for any interval $I \in \mathcal{I}$, we have*

$$|g(x)-g(y)| \le \frac{5M}{sL}, \quad x,y \in I, \tag{5.1}$$

where $M := \|g\|_{C^s[0,1]}$.

Proof: First, observe that $g(1)-g(0) =: M_0 \le M$. If $M_0 \le 5M/sL$, we have nothing to prove. So consider the case $M_0 \ge 5M/sL$. Let m be the smallest positive integer such that

$$\epsilon := \frac{M_0}{m} \le \frac{M}{sL}.$$

Then, $m \geq 5$ and also

$$M/\epsilon \geq sL \geq 2. \tag{5.2}$$

Now define the points $x_0 := 0 < x_1 < \ldots < x_m =: 1$ such that

$$g(x_i) - g(x_{i-1}) = \epsilon, \quad i = 1, \ldots, m.$$

From the assumption that $\|g\|_{C^s[0,1]} = M$, we have

$$\epsilon = |g(x_i) - g(x_{i-1})| \leq M|x_i - x_{i-1}|^s, \quad i = 1, \ldots, m.$$

It follows, therefore, that

$$|x_i - x_{i-1}| \geq (M/\epsilon)^{-1/s}, \quad i = 1, \ldots, m. \tag{5.3}$$

To prove (5.1), it is enough to show that each interval $I \in \mathcal{I}$ contains at most four of the points x_i, $i = 1, \ldots, m$. To do this, we shall now construct another adaptive partition. Given a pair (x_{i-1}, x_i), we consider the smallest sequence of dyadic subdivisions (starting with $[0,1]$) that are needed to separate these two points. If $\ell(x_{i-1}, x_i)$ is the number of such subdivisions, then from (5.3), we have $\ell(x_{i-1}, x_i) \leq \lceil s^{-1} \log_2(M/\epsilon) \rceil$. This means, we can find a dyadic partition $\mathcal{I}^* = \{I_1^*, \ldots, I_{\bar{n}}^*\}$ which simultaneously separates every such pair of these points by using at most

$$\bar{n} := \sum_{i=1}^{m} \ell(x_{i-1}, x_i) \leq m \lceil s^{-1} \log_2(M/\epsilon) \rceil \leq 2M_0 \epsilon^{-1} s^{-1} \log_2(M/\epsilon) \tag{5.4}$$

subdivisions. Here, we have used (5.2) to remove the $\lceil \cdot \rceil$. From the definition of m and the fact that $m \geq 2$, we have

$$\frac{M}{sL} \leq \frac{M_0}{m-1} \leq \frac{2M_0}{m} = 2\epsilon. \tag{5.5}$$

Hence, $M/\epsilon \leq 2sL \leq 2L$ and using that back in (5.4) gives

$$\bar{n} \leq 2M_0 s^{-1} \epsilon^{-1} \log_2(M/\epsilon) \leq 4M_0 s^{-1} \epsilon^{-1} \log_2 L \leq 8L \log_2 L \leq n. \tag{5.6}$$

Here in the second to last inequality we used the fact that $\log_2 2L \leq 2 \log_2 L$ for $L \geq 2$.

Finally, we observe that in the generation of the adaptive partition $\mathcal{I}$, if an interval generated in the subdivision process has 4 or more points x_i then it will always be subdivided before an interval with only one x_i. Since $\bar{n} \leq n$, no interval in $\mathcal{I}$ contains more than three point x_i. This means that the variation of g on any $I \in \mathcal{I}$ is less than 5ϵ and thus less than $5M/sL$. □

We shall also need the partition $\mathcal{J}$ of $[0,1]$ into L intervals of equal length. We now can describe the query points of our algorithm.

QMSTEP1: Evaluate g at points of $\mathcal{I}$. We ask for the values of f at the points corresponding to the endpoints of the intervals I_i of $\mathcal{I}$. The number of such query points is at most $n+1$ with $n = \lceil 8L \log_2 L \rceil + 1$.

QMSTEP2: Evaluate g at points of $\mathcal{J}$. We ask for the values of f at all of the endpoints of the intervals J_i of $\mathcal{J}$. The number of such query points is $L+1$.

QMSTEP3: Evaluate f at Bernoulli points. If $L < N$, we shall again using a Bernoulli matrix Φ of size $L \times N$. The entries in Φ are $\pm 1/\sqrt{L}$ and again we assume that we have a favorable draw so that the RIP and mapping properties introduced in §2 hold for the matrix Φ. We again denote by b_i, $i = 1, \dots, L$, the rows of Φ. Let us consider the points $z_i := 1/2 + \mu a \cdot b_i$, $i = 1, \dots, L$ where $\mu := \frac{\sqrt{L}}{2}$. All of these points are in $[0, 1]$. We can obtain the value of $g(z_i)$ by asking for the value of f at $(1/2, \dots, 1/2) + \mu b_i$. From this value, we can determine the interval $J \in \mathcal{J}$ which contains z_i. Now, we ask for the value of g at the midpoint of J. From this value, we can determine whether z_i is in the left or right child of J. Whichever child it is in, we ask for the value at its midpoint and proceed in the same way. Thus after $2\lceil \log_2 L \rceil$ of these steps, we will have an approximation $\hat{z}_i$ to z_i

$$|z_i - \hat{z}_i| \leq L^{-3}. \tag{5.7}$$

From this we obtain an approximation $\hat{y}_i$ to $y_i = a \cdot b_i$ satisfying

$$|y_i - \hat{y}_i| \leq 2L^{-7/2}. \tag{5.8}$$

We do this for each $i = 1, \dots, L$. Thus, this step will use $2L\lceil \log_2 L \rceil$ (adaptive) point queries of f.

If $L \geq N$, then as before we use the $N \times N$ identity matrix I_N in place of Φ and ask for the values of f at the coordinate points e_j which in turn gives $g(a_i)$. We determine which interval $J \in \mathcal{J}$ contains a_i and then do the adaptive subdivision as above to resolve a_i by $\hat{a}_i$ to accuracy

$$|a_i - \hat{a}_i| \leq L^{-3}. \tag{5.9}$$

Given these point values of f, we describe how we construct an approximation $\hat{f}$ to f.

RMSTEP1: Approximating g from the retrieved information. Using the values of g at the points of $\mathcal{I}$, we construct a piecewise linear interpolant $\hat{g}$ to g at these points. Notice that $\hat{g}$ is also monotone. Since g changes by at most $5M/sL$ between any two of these points, the function $\hat{g}$ satisfies

$$\|g - \hat{g}\|_{C[0,1]} \leq \frac{5M}{sL}. \tag{5.10}$$

RMSTEP2: Approximating a from the retrieved information. We first consider the case $L < N$. We apply the ℓ_1-minimization decoder Δ to the vector $\hat{y} := (\hat{y}_i)_{i=1}^L$. This gives $\tilde{a} := \Delta(\hat{y})$. Using the estimate (5.8), the same proof as in Lemma 3.1 shows that there is a constant C depending only on q such that

$$\|a - \tilde{a}\|_{\ell_1^N} \leq C\Big(M_1 \epsilon(N, L)^{1/q-1} + L^{-2}\Big). \tag{5.11}$$

As before, we modify $\tilde{a}$ to get $\hat{a}$ with positive entries and $\|\hat{a}\|_{\ell_1 N} = 1$. In the case $L \geq N$, we sum the estimates (5.9) to obtain

$$\|a - \hat{a}\|_{\ell_1^N} \leq L^{-2}, \tag{5.12}$$

which is the same form as (5.11).

With these results in hand, we obtain the following theorem.

Theorem 5.2 *The above algorithm uses at most $11L\lceil \log_2 L\rceil$ adaptive point queries. If $f(x) = g(a \cdot x)$ with g monotone, $a \in w\ell_q^N$, and $g \in C^s$ for some $0 < s \leq 1$, then the reconstruction from these queries gives an approximation $\hat{f}$ such that for all $L \geq 1$,*

$$\|f - \hat{f}\|_{C(\Omega)} \leq CM_0\Big(L^{-1} + C_1 M_1 \epsilon(N,L)^{s(1/q-1)}\Big), \tag{5.13}$$

where C is a constant depending only on s when s is small and C_1 depends only on q.

Proof: The proof is the same as the proof of Theorem 3.2 and so we do not repeat it here. □

6 Concluding remarks

While we have put forward results that show certain ridge functions in high dimension can be captured by a controlled number of queries, there are many directions and possible improvements that could be further explored.

6.1 The requirements on a

Our ultimate goal is to be able to treat general functions f of the form $f(x) = g(Ax)$ where A is an $m \times N$ matrix. This paper only discusses the simplest case of this where $m = 1$ in which case A is a vector a. In addition, we have imposed the requirement that the entries in a are nonnegative. While this matches some applications in econometrics (single index models), it would be desirable to have a theory that did not impose this requirement. We have used this assumption to guarantee that querying f at points of the form $t(1, \ldots, 1)$ gives the value of g at t. Obviously, we could also impose other specific sign patterns for the entries in a but this sign pattern must be known for the techniques of this paper to apply. It would be very desirable to develop techniques that weaken this assumption on a.

6.2 The optimality of our results

Our main result shows that with $\mathcal{O}(L)$ point value queries, any $f \in \mathcal{R}(s, q; M_0, M_1)$ can be retrieved with accuracy $M_0(L^{-s} + M_1\epsilon(N,L)^{1-1/q})$. It is legitimate to question the optimality of this result, in the sense that this order accuracy cannot be improved by *any* algorithm based on $\mathcal{O}(L)$ queries. Note that the algorithm proposed in this paper is *adaptive* in the sense that the points at which we call the values of f are not fixed in advance (except for the $L+1$ first base points), but rather depend on the information on f gained in previous queries. Therefore, any proof of optimality would have to allow all possible adaptive algorithms in the competition.

The following arguments show that our results are indeed optimal, save for the factor $\log_2(N/L)$ which appears in the $\epsilon(N,L)$.

In order to see that the term L^{-s} cannot be removed, we remark that for any set of L points, there exists a function $f \in \mathcal{R}(s, q; M_0, M_1)$ of the form

$$f(x) = g(x_1),$$

that vanishes at all these points and yet such that $\|f\|_{C(\Omega)} = \|g\|_{C([0,1])} \geq cM_0L^{-s}$, with $c > 0$ a constant that does not depend on L. Indeed there is an interval of length larger than $\frac{1}{2L}$ that does not contain any x_1 coordinate of the L points and we can take a function g compactly supported on this interval such that $\|g\|_{C^s} \leq M_0$ and $\|g\|_{C([0,1])} \geq cM_0 L^{-s}$. We now challenge any adaptive algorithm by considering the sequence of points which are queried by this algorithm when all measured values are zero. For this sequence, the corresponding function f constructed as above gives the same values as $-f$, and in turn the reconstruction error over the class $\mathcal{R}(s, q; M_0, M_1)$ can be as bad as $cM_0 L^{-s}$.

In order to show that the term $\epsilon(N, L)^{1-1/q}$ cannot be removed, a possibility is to consider functions in $\mathcal{R}(s, q; M_0, M_1)$ of the form

$$f(x) = M_0 a \cdot x,$$

with $a \geq 0$ and $\|a\|_{w\ell_q^N} \leq M_1$, or equivalently

$$f(x) = b \cdot x,$$

with $b \geq 0$ and $\|b\|_{w\ell_q^N} \leq M_0 M_1$. Recent results on Gelfand widths [10] show that for any sequence of L vectors $x^{(1)}, \cdots, x^{(L)}$, there exists a vector b orthogonal to all $x^{(i)}$ such that $\|b\|_{w\ell_q^N} \leq M_0 M_1$ and

$$\|b\|_{\ell_1^N} \geq c\varepsilon(N, L)^{1-1/q},$$

with $c > 0$ a constant that does not depend on L and N. By considering the functions f and $-f$, this would be sufficient to derive the optimality of the term $\epsilon(N, L)^{1-1/q}$ by the same argument as for the term L^{-s}, if we had not imposed that the vector b should have positive coordinates.

For this reason, the optimality of the term $\epsilon(N, L)^{1-1/q}$ over $\mathcal{R}(s, q; M_0, M_1)$ for all adaptive algorithms remains unclear to us. It may be proved if we only challenge non-adaptive algorithms (but then our algorithm does not fall in this category), or if we give up on the factor $\log_2(N/L)$ which appears in the $\epsilon(N, L)$. We explain this second option.

We consider the function $f(x) = b \cdot x$ with vector

$$b = M_0 M_1 2^{-2/q}(L^{-1/q}, \cdots, L^{-1/q}, 0, \cdots, 0),$$

where the first $2L$ coordinates are non-zero. Then, $\|b\|_{w\ell_q^N}^q \leq \|b\|_{\ell_q^N}^q \leq 1/2$. For any adaptive algorithm, we consider the sequence $x^{(1)}, \cdots, x^{(L)}$ that is picked for this particular function. Then, classical results on Gelfand width of hypercubes (see [11] Theorem 3.2, p.410) show that there exists a vector b', with only its first $2L$ coordinates nonzero, which is orthogonal to all $x^{(i)}$ and such that

$$\|b'\|_{\ell_\infty} \leq 1 \text{ and } \|b'\|_{\ell_1} \geq L.$$

We then define $\tilde{b} = b + M_0 M_1 2^{-2/q} L^{-1/q} b'$, so that $\|\tilde{b}\|_{w\ell_q^N} \leq 1$ and

$$\|b - \tilde{b}\|_{\ell_1} \geq M_0 M_1 2^{-2/q} L^{1-1/q}.$$

We set $\tilde{f}(x) = \tilde{b} \cdot x$. Both functions f and $\tilde{f}$ now belong to $\mathcal{R}(s, q; M_0, M_1)$ and give the same values. It is now easy to check that

$$\|f - \tilde{f}\|_{C([0,1])} \geq M_0 M_1 2^{-2/q-1} L^{1-1/q}.$$

6.3 Comparison with [8]

It is interesting to compare the above approximation result with those that could be derived if we were following the approach proposed in [8]. The latter allows us to recover any function f that differs by at most τ from a C^s function of k unknown variables, to an accuracy

$$\|f - \hat{f}\|_{C(\Omega)} \leq C(L^{-s/k} + \tau),$$

using $\mathcal{O}(L \log N)$ queries of f, where the constant C depends on k. Since a can be approximated in ℓ_1 by a vector with k non-zero coordinates to an accuracy of order $k^{1-1/q}$, this implies that f differs from a function of k unknown variables by an error of order $\tau = k^{1-1/q}$. Therefore, application of the approach proposed in [8] results in the error bound

$$\|f - \hat{f}\|_{C(\Omega)} \leq C\Big(L^{-s/k} + k^{1-1/q}\Big), \tag{6.1}$$

for any $k > 0$, where C depends on k, which is less favorable than our results since we need to make k large and this deteriorates the rate in L.

On the other hand, in contrast to [8], the algorithm given in this paper is less robust with respect to a deviation of f from the model class of ridge functions, as noticed in §4.

References

[1] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, *A simple proof of the restricted isometry property for random matrices*, Constructive Approximation, **28**(2008), 253–263.

[2] E. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure and Appl. Math., **59**(2006), 1207–1223.

[3] A. Cohen, W. Dahmen, and R. DeVore, *Compressed sensing and best k term approximation*, JAMS **22**(2009), 211–231.

[4] A. Cohen, R. DeVore, C. Schwab, *Convergence rates of best n term Galerkin approximations for a class of elliptic SPDEs*, J. FOCM, to appear.

[5] R. Coifman and M. Maggioni, *Diffusion wavelets*, Appl. Comp. Harm. Anal., **21**(1) (2006), 53–94.

[6] R. DeVore, *Nonlinear Approximation*, Acta Numerica, Volume 7 (1998), 51-150.

[7] R. DeVore and G.G. Lorentz, *Constructive Approximation*, vol. 303, Grundlehren, Springer Verlag, N.Y., 1993.

[8] R. DeVore, G. Petrova, and P. Wojtaszczyk, *Approximating functions of few variables in high dimensions* , Constructive Approximation, to appear

[9] R. DeVore, G. Petrova, and P. Wojtaszczyk, Instance optimality in probability with an ℓ_1-minimization decoder, Appl. Comput. Harmon. Anal., **27**(2009), 275–288.

[10] S. Foucart, A. Pajor, H. Rauhut, T. Ullrich, *The Gelfand widths of ℓ_p balls for* $0 < p \leq 1$, preprint.

[11] G.G. Lorentz, M. Von Golitschek and Y. Makovoz, *Constructive Approximation - Advances Problems*, vol. 304, Grundlehren, Springer Verlag, N.Y., 1996.

[12] M.H. Maathuis, M. Kalisch and P. Buhlmann, *Estimating high-dimensional intervention effects from observational data*, Annals of Statistics Annals of Statistics **37**(2009), 3133–3164.

[13] M. Belkin and P. Niyogi, *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, Neural Computation, **15** (2003), 1373–1396.

[14] E. Novak and H. Woźniakowski, *Tractability of Multivariate Problems vol. I: Linear Information*, European Math. Soc., 2008.

[15] J. Haupt, R. Castro and R. Nowak, *Distilled Sensing: Adaptive Sampling for Sparse Detection and Estimation*, preprint, 2010.

[16] C. J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13:689–705, 1985.

[17] S. Gaiffas and G. Lecue. Optimal rates and adaptation in the single-index model using aggregation. *Electronic Journal of Statistics*, 1:538, 2007.

[18] A. Juditsky, O. Lepski, and A. Tsybakov. Nonparametric estimation of composite functions. *Ann. Stat.*, 37(3):1360–1404, June 2009.

[19] G. K. Golubev. Asymptotically minimax estimation of a regression function in an additive model. *Problemy Peredachi Informatsii*, 28:101–112, 1992.

[20] Z. Chi. *On ℓ_1-regularized estimation for nonlinear models that have sparse underlying linear structures.* ArXiv e-prints, Nov. 2009.

[21] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[22] J. Traub and H. Wozniakowski, A General Theory of Optimal Algorithms, Academic Press, New York, 1980.

[23] M. J. Wainwright, *Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting*, IEEE Trans. Inf. Theory, **55**(2009), 5728-5741.

[24] J. Traub, G. Wassilkowski and H. Wozniakowski, Information-Based Complexity, Academic Press, New York, NY, 1988.

Albert Cohen, Université Pierre et Marie Curie, Laboratoire Jacques-Louis Lions, Paris, France, cohen@ann.jussieu.fr

Ingrid Daubechies, Department of Mathematics, Princeton University, Princeton, NJ, USA
ingrid@math.princeton.edu

Ronald DeVore, Department of Mathematics, Texas A&M University, College Station, TX, USA
rdevore@math.tamu.edu

Gerard Kerkyacharian, Université Paris Diderot. Laboratoire PMA. F-75013, Paris, France, kerk@math.jussieu.fr

Dominique Picard, Université Paris Diderot. Laboratoire PMA. F-75013, Paris, France, picard@math.jussieu.fr