Nonlinear approximation and its applications

Ronald A. DeVore

Abstract I first met Wolfgang Dahmen in 1974 in Oberwolfach. He looked like a high school student to me but he impressed everyone with his talk on whether polynomial operators could produce both polynomial and spectral orders of approximation. We became the best of friends and frequent collaborators. While Wolfgang's mathematical contributions spread across many disciplines, a major thread in his work has been the exploitation of nonlinear approximation. This article will reflect on Wolfgang's pervasive contributions to the development of nonlinear approximation and its application. Since many of the contributions in this volume will address specific application areas in some details, my thoughts on these will be to a large extent anecdotal.

1 The early years

I was first exposed to approximation theory in a class taught by Ranko Bojanic in the Fall of 1964 at Ohio State University. Each student was allowed one optional class (outside of the required algebra and analysis). I do not know why I chose this from among the other options - perhaps another student had recommended it to me as a well structured interesting class - but I was immediately hooked. It just seemed like a natural subject answering natural questions. If we cannot explicitly solve most real world problems then we better learn how to approximate them.

The course was more on the theory than on the computational side since the demand for fast computational algorithms did not yet seem as urgent. There were no wavelets and splines were in their infancy. But there was plenty to intrigue the student including the Jackson-Bernstein theory of polynomial approximation which remains to this day as the prototype for understanding the quantitative side of ap-

Ronald A. DeVore
Texas A& M University, College Station, TX, USA, e-mail: ronald.a.devore@gmail.com

R.A. DeVore, A. Kunoth (eds.), *Multiscale, Nonlinear and Adaptive Approximation*,
DOI 10.1007/978-3-642-03413-8_6,

proximation. Let us describe the modern form of this theory since it will be useful as we continue this exposition.

Suppose that we are interested in approximating the elements from a space X equipped with a norm $\|\cdot\| := \|\cdot\|_X$ by using the elements of the spaces X_n, $n = 1,2,\ldots$. Typical examples are $X = L_p$ or a Sobolev space while the usual suspects for X_n are spaces of polynomials, splines, or rational functions. We assume that for all $n,m \geq 1$, we have

$$X_n + X_m \subset X_{c(n+m)}, \text{ for some fixed } c \geq 1, \tag{1}$$

which is certainly the case for the above examples. Given $f \in X$, we define

$$E_n(f) := \inf_{g \in X_n} \|f - g\|. \tag{2}$$

The main challenge in the quantitative arena of approximation is to describe precisely the elements of X which have a prescribed order of approximation. Special attention is given to the approximation orders which are of the form n^{-r} since these occur most often in numerical computation. This gives the *primary approximation spaces* $\mathscr{A}^r := \mathscr{A}^r(X,(X_n))$, $r > 0$, consisting of all $f \in X$ for which

$$|f|_{\mathscr{A}^r} := \sup_{n \geq 1} n^r E_n(f) \tag{3}$$

is finite. The left side of (3) serves to define a semi-norm on $\mathscr{A}^r$. We obtain the norm for this space by adding $\|f\|_X$ to the semi-norm.

While the spaces $\mathscr{A}^r$ are sufficient to understand most approximation methods, it is sometimes necessary to go to a finer scale of spaces when dealing with nonlinear approximation. Accordingly, if $q > 0$, we define $\mathscr{A}^r_q$ via the quasi-norm

$$|f|_{\mathscr{A}^r_q(X)} := \|(2^{kr} E_{2^k}(f))\|_{\ell_q}. \tag{4}$$

Again, we obtain the norm for this space by adding $\|f\|_X$ to the semi-norm. When $q = \infty$, we obtain the spaces $\mathscr{A}^r$ because of (1).

The problem of characterizing $\mathscr{A}^r$ was treated in the following way for the case when $X = C[-\pi,\pi]$ is the space of continuous 2π periodic functions and X_n is the space of trigonometric polynomials of degree $\leq n$. One proves two fundamental inequalities for trigonometric approximation. The first of these is the following inequality proved by D. Jackson:

$$E_n(f) \leq C_k \|f^{(k)}\|_{C[-\pi,\pi]} n^{-k}, \quad n,k = 1,2,\ldots. \tag{5}$$

A companion to this is the famous Bernstein inequality which says

$$\|T^{(k)}\|_{C[-\pi,\pi]} \leq n^k \|T\|_{C[-\pi,\pi]}, \quad n,k = 1,2,\ldots \tag{6}$$

From these two fundamental inequalities, one can show that $\mathscr{A}^r$ is the generalized Lipschitz space Lip r space (defined later in §2) and more generally the $\mathscr{A}^r_q$ are

the same as the Besov spaces $B_q^r(L_\infty)$ which are also discussed in §2. It is interesting to note that the modern way of deriving such characterizations is not much different than the classical approach for trigonometric polynomials except that everything is now encasted in the general framework of interpolations spaces. This leads to the following theory.

Suppose for our approximation setting, we can find a space Y_k such that the following generalized Jackson and Bernstein inequalities hold

$$E_n(f)_X \le C_k \|f\|_{Y_k} n^{-k}, \quad n = 1,2,\dots. \tag{7}$$

and

$$\|S\|_{Y_k} \le C_k n^k \|S\|_X, \quad S \in X_n, \quad n = 1,2,\dots. \tag{8}$$

Then for any $0 < r < k$ and $0 < q \le \infty$, we have

$$\mathscr{A}_q^r(X,(X_n)) = (X,Y_k)_{\theta,q}, \quad \theta := r/k, \tag{9}$$

where the spaces on the right are the interpolation spaces given by the real method of interpolation (K-functionals) as described in the next section. In our case of trigonometric polynomial approximation the space Y_k is C^k with its usual semi-norm. It is well known that the interpolation spaces between C and C^k are the Besov spaces and in particular the generalized Lipschitz spaces when $q = \infty$.

The beauty of the above theory is that it boils down the problem of characterizing the approximation spaces for a given method of approximation to one of proving two inequalities: the generalized Jackson and Bernstein inequalities for the given approximation process. This recipe has been followed many times. An interesting question is whether the characterization (9) provides essential new information. That this is indeed the case rests on the fact that these interpolation spaces can be given a concrete description for most pairs (X,Y_k) of interest. This fact will be discussed next.

2 Smoothness and interpolation spaces

We all learn early on that the more derivatives a function has then the smoother it is. This is the coarse idea of smoothness spaces. Modern analysis carries this idea extensively forward by introducing a myriad of spaces to delineate properties of functions. We will touch on this with a very broad stroke only to communicate the heuerestic idea behind the smoothness spaces we shall need for describing rates of approximation.

For an integer $s > 0$, the Sobolev space $W^s(L_p(\Omega))$, on a domain $\Omega \subset \mathbb{R}^d$ consists of all $f \in L_p(\Omega)$ for which all of the distributional derivatives $D^\nu f$ of order s are also in $L_p(\Omega)$. This space is equipped with the semi-norm

$$|f|_{W^s(L_p(\Omega))} := \max_{|\nu|=s} \|D^\nu f\|_{L_p(\Omega)}. \tag{10}$$

We obtain the norm on $W^s(L_p(\Omega))$ by adding $\|\cdot\|_{L_p(\Omega)}$ to this semi-norm.

It is of great interest to extend this definition to all $s > 0$. One can initiate such an extension from many viewpoints. But the most robust of these approaches is to replace derivatives by differences. Suppose that we wish to define fractional order smoothness spaces on $\mathbb{R}^d$. The translation operator T_h for $h \in \mathbb{R}^d$ is defined on a function f by $T_h(f) := f(\cdot + h)$ and leads to the difference operators

$$\Delta_h^r := \sum_{k=0}^{r} (-1)^{r-k} \binom{r}{k} T_{kh}. \tag{11}$$

If we apply Δ_h^r to a smooth function f then $h^{-r}\Delta_h^r(f)(x) \to r! f^{(r)}(x)$ as $h \to 0$. We can obtain smoothness spaces in L_p by placing conditions on how fast $\|\Delta_h^r(f)\|_{L_p}$ tends to zero as $h \to 0$. To measure this we introduce the *moduli of smoothness*

$$\omega_r(f,t)_p := \sup_{|h| \le t} \|\Delta_h^r(f)\|_{L_p(\Omega_{rh})}, \tag{12}$$

where Ω_t consists of all $x \in \Omega$ for which the line segment $[x, x+t]$ is contained in Ω.

We get a variety of spaces by placing decay conditions on $\omega_r(f,t)_p$ as $t \to 0$. The most classical of these are the generalized Lipschitz spaces $\mathrm{Lip}\,\alpha := \mathrm{Lip}(\alpha, L_p)$ in L_p which consist of all f for which

$$|f|_{\mathrm{Lip}(\alpha,L_p)} := \sup_{t>0} t^{-\alpha} \omega_r(f,t)_p, \quad \alpha < r, \tag{13}$$

is finite. We obtain the norm on this space by adding $\|f\|_{L_p}$ to (13). The above definition holds for all $0 < p \le \infty$. We usually make the convention that L_∞ is replaced by the space of continuous functions. Note that the above definition apparently depends on r but it is easy to show that one obtains exactly the same spaces no matter which r one choses (as long as $r > \alpha$) and the (quasi-)seminorms (13) are equivalent.

The generalized Lipschitz spaces are fine for a good understanding of approximation. However, certain subtle questions require a finer scaling of spaces provided by the Besov scale. Now, in addition to α we introduce a second fine scale parameter $q \in [0, \infty)$. Then the Besov space $B_q^\alpha(L_p)$ is defined by its semi-norm

$$|f|_{B_q^\alpha(L_p)} := \{ \int_{t>0} [t^{-\alpha} \omega_r(f,t)_p]^q \frac{dt}{t} \}^{1/q}, \quad \alpha < r. \tag{14}$$

2.1 The role of interpolation

We have already noted that approximation spaces can be characterized as interpolation spaces provided the fundamental Bernstein and Jackson type inequalities have been proven. For this characterization to be of use, we need to be able to describe

these interpolation spaces. Although this is not always simple, it has been carried out for all pairs of spaces that arise in linear and nonlinear approximation. To describe these results we will make a very brief incursion into interpolation.

The subject of operator interpolation grew out of harmonic analysis in the quest to have a unified approach to characterizing the mapping properties of its primary operators such as Fourier transforms, conjugate operators, maximal functions and singular integrals. Of primary interest in approximation theory are the real interpolation spaces. Given a pair of normed linear spaces X, Y which are both embedded in a common topological space, we can define the K-functional

$$K(f,t) := K(f,t;X,Y) := \inf_{f=f_0+f_1} \{\|f_0\|_X + t\|f_1\|_Y\}. \tag{15}$$

Often, the norm on Y is replaced by a semi-norm as is the case below when considering Y as a Sobolev space. The real interpolation spaces $(X,Y)_{\theta,q}$ are now defined for any $\theta \in (0,1)$ and $q > 0$ by the quasi-norm

$$\|f\|_{(X,Y)_{\theta,q}} := \|t^{-\theta} K(f,t)\|_{L_q(\mu)}, \tag{16}$$

where $\mu(t) := \frac{dt}{t}$ is Haar measure. By this time the reader is sure to observe the common flavor of all these norms (approximation spaces, Besov spaces, and interpolation spaces).

We have already mentioned that these interpolation spaces are identical to the approximation spaces whenever we have the Jackson and Bernstein inequalities in fold. What is ever more enlightening is that for classical pairs of spaces the K-functional and the interpolation spaces are always familiar quantities which have been walking the streets of analysis for decades. Let us give a couple of examples which will certainly convince even the most skeptical reader of the beautiful way in which the whole story pieces together.

The L_p spaces are interpolation spaces for the pair (L_1, L_∞) as is encapsulated in the Riesz-Thorin interpolation theorem (usually proved by means of complex interpolation). This theorem also follows from the real method of interpolation since the K-functional for this pair is easy to describe

$$K(f,t,L_1,L_\infty) = \int_0^t f^*(s)\,ds, \tag{17}$$

where f^* is the nondecreasing rearrangement of f as introduced by Hardy and Littlewood. From this characterization, one easily deduces that the interpolation spaces $(L_1,L_\infty)_{\theta,q}$ are identical to the Lorentz spaces $L_{p,q}$ with the identification $\theta = 1 - 1/p$. When $q = p$, we obtain $L_p = L_{p,p}$.

As a second example, consider the K-functional for the pair $(L_p(\Omega), W^k(L_p(\Omega)))$ on a Lipschitz domain $\Omega \subset \mathbb{R}^d$. Johnen and Scherer [37] showed that

$$K(f,t,L_p(\Omega),W^k(L_p(\Omega)) \approx \omega_r(f,t)_p \tag{18}$$

our old friend the modulus of smoothness. From this, one immediately deduces that $(L_p(\Omega), W^k(L_p(\Omega))_{\theta,q} = B^s_q(L_p(\Omega))$ for $\theta = s/k$.

There are numerous other examples of this sort beautifully reported on in the book by Bennett and Sharpley [8] that unquestionably convince us that the K-functional is indeed a natural object. These results make our job of characterizing the approximation spaces quite clear. We need only establish corresponding Jackson and Bernstein inequalities for the given approximation process and then finish the characterization via interpolation theory. This will be our modus operandi in the sequel.

3 The main types of nonlinear approximation

In application domains, there are four types of nonlinear approximation that are dominant. We want to see what form the general theory takes for these cases. We suppose that we are interested in approximating the elements $f \in X$ where X is a (quasi-) Banach space equipped with a norm $\|\cdot\|_X$.

3.1 n-term approximation

A set $\mathscr{D} \subset X$ of elements from X is called a dictionary if each element $g \in \mathscr{D}$ has norm one and the finite linear combinations of the elements in $\mathscr{D}$ are dense in X. The simplest example of a dictionary is when $\mathscr{D}$ is a basis for X. However, redundant systems $\mathscr{D}$ are also important. An issue is how much redundancy is possible while retaining reasonable computation.

Given a positive integer n, we define Σ_n as the set of all linear combinations of at most n elements from $\mathscr{D}$. Thus, the general element in Σ_n takes the form

$$S = \sum_{g\in\Lambda} c_g g, \quad \#(\Lambda) = n. \tag{19}$$

If we use the elements of Σ_n to approximate a function $f \in X$, then it induces an error

$$\sigma_n(f)_X := \inf_{S\in\Sigma_n} \|f - S\|_X. \tag{20}$$

Here we are following tradition to denote the error of nonlinear approximation by σ_n rather than using the generic E_n introduced earlier. The approximation spaces $\mathscr{A}^r_q(X)$ are defined as in the general setting. The approximation problem before us is whether we can characterize these spaces.

Let us consider the simplest case of the above setting where $X = \mathscr{H}$ is a real Hilbert space and $\mathscr{D} = \{\phi_j\}_{j=1}^\infty$ is an orthonormal basis for $\mathscr{H}$. Then, each $f \in \mathscr{H}$ has an orthogonal expansion

$$f = \sum_{j=1}^{\infty} \langle f, \phi_j \rangle \phi_j, \quad \|f\|_{\mathscr{H}}^2 = \sum_{j=1}^{\infty} \langle f, \phi_j \rangle^2. \tag{21}$$

Because of the $\mathscr{H} \longrightarrow \ell_2$ isometry, a best n term approximation to a given $f \in \mathscr{H}$ is obtained by retaining its n largest terms (the possibility of ties in the size of the coefficients shows that this best approximation is not necessarily unique). Thus, if we let $c_j = \langle f, \phi_j \rangle$ and (c_j^*) be the rearrangement of their absolute values into nonincreasing order, then the approximation error of n-term approximation to f is

$$\sigma_n^2(f) = \sum_{j>n} [c_j^*]^2, \quad n = 1, 2, \ldots. \tag{22}$$

There is a simple characterization of the approximation spaces in this setting of n-term approximation. For example, for the primary spaces, we have that $f \in \mathscr{A}^r$ if and only if the coefficients (c_j) are in weak ℓ_τ with $1/\tau = s + 1/2$ and

$$\|f\|_{\mathscr{A}^r} \approx \|(c_j)\|_{w\ell_\tau}, \tag{23}$$

where we recall that weak ℓ_τ is the space of all sequences (a_j) which satisfy

$$\|(a_j)\|_{w\ell_\tau} := \sup_{n \geq 1} n^{1/\tau} a_n^* < \infty. \tag{24}$$

Similar results hold for the secondary spaces $\mathscr{A}_q^r$ characterizing them by the membership of the coefficient sequences in the Lorentz spaces $\ell_{\tau,q}$, $1/\tau = s + 1/2$. Indeed, this can be proved by establishing generalized Jackson-Bernstein inequalities for the pair $\mathscr{H}$ and Y_k as the set of $f \in \mathscr{H}$ whose coefficient are in weak ℓ_p with $1/p = k + 1/2$. We refer the reader to [25] for details.

In the case where we are interested in approximation in other spaces than $\mathscr{H}$, for example in L_p, $p \neq 2$, things are more subtle and depend very much on the particular basis $\{\phi_j\}$. Let us restrict our attention to to the wavelet basis which will play a special role in our discussion.

Suppose that φ is a compactly supported univariate scaling function (i.e. φ satisfies a two scale relationship) whose shifts form an orthonormal system. Let ψ be the compactly supported mother wavelet associated to φ normalized in $L_2(\mathbb{R}^d)$: $\|\psi\|_{L_2} = 1$. There are two ways to form an orthonormal wavelet system from this pair. The standard construction is to define $\psi^0 := \varphi$ and $\psi^1 := \psi$. If E' is the set of vertices of the unit cube and E the set of nonzero vertices, we define

$$\psi^e(x_1, \ldots, x_d) := \psi^{e_1}(x_1) \cdots \psi^{e_d}(x_d), \quad e \in E'. \tag{25}$$

The shifted dilates $\psi_{j,k}^e(x) := 2^{jd/2} \psi^e(2^j(x-k))$, $j \in \mathbb{Z}$, $k \in \mathbb{Z}^d$, $e \in E$, form an orthonormal system for $L_2(\mathbb{R}^d)$.

It is convenient to index these wavelets according to their spacial scaling. Let $\mathscr{D}(\mathbb{R}^d)$ denote the set of all dyadic cubes in $\mathbb{R}^d$. Each $I \in \mathscr{D}(\mathbb{R}^d)$ has the form $I = 2^{-jd}[k, k + \underline{1}]$ with $\underline{1} := (1, \ldots, 1)$. We identify the wavelets with the dyadic cubes via

$$\psi_I^e := \psi_{j,k}^e, \quad I \in \mathscr{D}(\mathbb{R}^d), e \in E. \tag{26}$$

This gives the wavelet decomposition

$$f = \sum_{I \in \mathscr{D}} \sum_{e \in E} f_{I,e} \psi_I^e, \quad f_{I,e} := \langle f, \psi_I^e \rangle, \tag{27}$$

which is valid for each $f \in L_1(\mathbb{R}^d) + L_\infty(\mathbb{R}^d)$.

There is a second wavelet basis built directly from tensor products of univariate wavelets. If $R = I_1 \times \cdots \times I_d$, $I_j \in \mathscr{D}(\mathbb{R}^d)$, $j = 1, \ldots, d$, is a d dimensional dyadic rectangle, then we define

$$\psi_R(x) := \psi_{I_1}(x_1) \cdots \psi_{I_d}(x_d), \tag{28}$$

where each ψ_{I_j} is a univariate wavelet. This basis is sometimes called the hyperbolic wavelet basis or sparse grid basis in PDEs. The support of ψ_R is now associated to the rectangle R and in the case that ψ is the univariate Haar wavelet it is precisely this rectangle.

To continue the discussion, let us consider the first of these bases. Some of the results for L_2 approximation carry over to other approximation norms. The vehicle for doing this is the Littlewood-Paley theory for wavelets which allows one to compute other norms such as the L_p norms by simple expressions (the square function) of the wavelet coefficients. Rather than go too far down this road, which is well reported on in [25], we mention only some of the consequences of this. The first of which is the fact that it is possible to characterize the approximation spaces $\mathscr{A}_q^r(L_p)$ for certain special values of q even when the approximation takes place in an L_p space, $p > 1$. This even extends to $p \leq 1$ if we replace the L_p space by the Hardy space H_p. Namely, $\mathscr{A}_q^r(L_p(\mathbb{R}^d)) = B_q^{rd}(L_q(\mathbb{R}^d))$, provided $1/q = r + 1/p$. These results carry over to approximation on domains $\Omega \subset \mathbb{R}^d$ but now more care must be taken to define appropriate wavelet bases. The only case that is completely straightforward is to use the Haar wavelets for a cube such as $[0,1]^d$ in $\mathbb{R}^d$.

From the Besov characterizations of the approximation spaces given in the previous paragraph, we can see the power of nonlinear approximation. If we use the elements from linear spaces of dimension n (such as polynomials or splines on uniform partitions) to approximate a function $f \in L_p(\Omega)$, $\Omega \subset \mathbb{R}^d$, then we will obtain approximation of order $O(n^{-r})$ if and only if $f \in B_\infty^{rd}(L_p(\Omega)$, i.e. roughly speaking we need f to have rd derivatives in L_p. However, when using nonlinear methods such as n-term wavelet approximation it is sufficient to have $f \in B_q^{rd}(L_q)$, $1/q = r + 1/p$, i.e. rd derivatives in L_q. The gain here is not in the number of derivatives (rd) but in the space where these derivatives must lie. Since $q < p$ this requirement is much weaker in the case of nonlinear approximation. Indeed, functions with singularities may be in $f \in B_q^{rd}(L_q)$ but not in $f \in B_\infty^{rd}(L_p)$.

Here is a useful way to think about this comparison between linear and nonlinear for approximation in L_p. If we use linear methods, there will be a largest value s_L such that $f \in B_\infty^s(L_p)$ for all $s < s_L$. Similarly, there will be a largest s_{NL} such that $f \in B_q^s(L_q)$, $1/q = s/d + 1/p$ for all $s < s_{NL}$. We always have $s_{NL} \geq s_L$. However,

in many cases s_{NL} is much larger than s_L. This translates into being able to approximate such f with accuracy $O(n^{-s_{NL}/d})$ for nonlinear methods with n parameters but only accuracy $O(n^{-s_L/d})$ for linear methods with the same number of parameters. Consider the case $d = 1$ and a function f which is piecewise analytic with a finite number of jump discontinuities. If we approximate this function in $L_2[0,1]$ using linear spaces of dimension n, we will never get approximation orders better than $O(n^{-1/2})$ because $s_L = 1/2$, but using nonlinear methods we obtain order $O(n^{-r})$ for all $r > 0$ because $s_{NL} = \infty$.

Let us turn to the question of how we build a good n-term approximation to a function $f \in L_p$ where there is an important story to tell. It is very simple to describe how to choose a near best n-term approximation to a given f by simply choosing the n-terms in the wavelet expansion for which $\|f_{I,e}\psi_I^e\|_{L_p}$ is largest. Let $\tilde{\Lambda}_n(f) := \{(I,e)\}$ be the indices of these n largest terms (with ties in the size of the coefficients handled in an arbitrary way) and $S_n(f) := \sum_{(I,e)\in\tilde{\Lambda}_n(f)} f_{I,e}\psi_I^e$. Then we have the beautiful result of Temlyakov[50]

$$\|f - S_n(f)\|_{L_p(\mathbb{R}^d)} \le C\sigma_n(f)_{L_p(\mathbb{R}^d)}, \tag{29}$$

with the constant C depending only on d and p.

Sometimes it is notationally beneficial to renormalize the wavelets in L_p. Let us denote by $\psi_{I,p}^e$ these renormalized wavelets and by $f_{I,e,p}$ the coefficients of f with respect to this renormalized bases. Then a consequence of (29) is that a simple thresholding of the wavelet coefficients yields near best approximants. Namely, given any threshold $\delta > 0$, we denote by $\Lambda_\delta(f) := \Lambda_{\delta,p}(f) := \{(I,e) : |f_{I,e,p}| > \delta\}$, and the approximation

$$T_\delta(f) := \sum_{(I,e)\in\Lambda_\delta(f)} f_{I,e,p}\psi_{I,p}^e. \tag{30}$$

Then, $T_\delta(f)$ is a near best n-term approximation to f in $L_p(\mathbb{R}^d)$ for $n = \#(\Lambda_\delta(f))$. Notice that there is a slight distinction here between $T_\delta(f)$ and $S_n(f)$ because for some values of n, $S_n(f)$ cannot be obtained by thresholding because of possible ties in the size of coefficients.

Let us conclude this discussion of n-term approximation by remarking that it cannot be implemented directly in a numerical application because it requires a search over all wavelet coefficients which is an infinite task. In numerical practice this search is limited by fixing a maximal dyadic level J to limit the search. Other numerically friendly nonlinear algorithms are adaptive and tree based algorithm which we discuss next.

3.2 Adaptive approximation

This type of approximation has a long history and owes a lot of its interest to its usefulness in describing certain numerical methods for PDEs. To drive home the main ideas behind adaptive approximation, let us consider the simple setting of approximating a function f on the unit cube $\Omega := [0,1]^d$ in $\mathbb{R}^d$ using piecewise polynomials on partitions consisting of dyadic cubes from $\mathscr{D}(\Omega) := \{I \in \mathscr{D}(\mathbb{R}^d) : I \subset \Omega\}$. Given an integer $r > 0$ and an $f \in L_p(\Omega)$, we denote by

$$E_r(f,I)_p := \inf_{Q \in \mathscr{P}_{r-1}} \|f - Q\|_{L_p(I)}, \tag{31}$$

the L_p error in approximating f on I by polynomials of order r (total degree $r-1$). The simplest adaptive algorithms are built on an estimator $E(I)$ for $E_r(f,I)_p$:

$$E_r(f,I)_p \le E(I), \quad I \in \mathscr{D}(\Omega). \tag{32}$$

To build an adaptive approximation to f, we let $\Lambda_0 := \{\Omega)\}$ and given that $\Lambda_n = \Lambda_n(f)$ has been defined, we generate Λ_{n+1} by choosing the dyadic cube $I = I_n$ from Λ_n for which the estimator $E(I_n)$ is largest (with again ties handled arbitrarily) and then removing I and replacing it by its 2^d children. Thus, the idea is to only subdivide where the error is largest. There have been several papers discussing the approximation properties of such adaptive algorithms starting with the pioneering work of Birman and Solomjak [13] which established convergence rates (in the case $E(I) = E(f,I)_p$) very similar to the estimates of the previous section for n-term wavelet approximation. A typical result is that if a function f is in a Besov space $B^s_q(L_\tau)$ which compactly embeds into L_p then a suitable adaptive algorithm will provide an approximation to f with accuracy $O(n^{-s/d})$ where n is the number of parameters (proportional to the number of cells in the adaptive partition). One can easily argue that one cannot do away with the assumption of compact embedding. Such results on adaptive approximation are only slightly weaker than those for n-term approximation. In the latter one does not assume compactness of the embedding into L_p.

One can even guarantee a certain near optimal performance of adaptive algorithms although now the rule for subdividing is more subtle. These will be described in the next section in the more general setting of tree approximation.

3.3 Tree approximation

We have already noted that trees arise in a natural way in nonliner approximation. The wavelet decomposition organizes itself on trees whose nodes are dyadic cubes in $\mathbb{R}^d$. We have also seen that adaptive partitioning is described by a tree whose nodes are the cells created during the adaptive algorithm. It is useful to formalize

tree approximation and extract its main features since we shall see that it plays a significant role in applications of nonlinear approximation.

We assume that we have a (generally infinite) master tree $\mathscr{T}^*$ with one root node. In the case of adaptive partitioning this root node would be the domain Ω. We also assume that each node has exactly K children. This matches both the wavelet tree and the usual refinement rules in adaptive partitioning. Note that in the case the master tree arises from adaptive partitioning, it fixes the way a cell must be subdivided when it arises in an adaptive algorithm. So this setting does not necessarily cover all possible adaptive strategies.

We shall be interested in finite subtrees $\mathscr{T} \subset \mathscr{T}^*$. Such a tree $\mathscr{T}$ has the property that for any node in $\mathscr{T}$ its parent is also in $\mathscr{T}$. We define $\mathscr{L}(T)$ to be the leaves of $\mathscr{T}$. This is the set of all terminal nodes in $\mathscr{T}$, i.e. such a node has none of its children in $\mathscr{T}$. We say that the tree is complete if whenever a node is in $\mathscr{T}$ all of its siblings are also in $\mathscr{T}$. We shall restrict our discussion to complete trees. Any adaptively generated partition is associated to a complete tree $\mathscr{T}$. We define $\mathscr{N}(\mathscr{T})$ to be the set of the internal nodes of $\mathscr{T}$, i.e. the ones which are not leaves. Then $\mathscr{T} = \mathscr{N}(\mathscr{T}) \cup \mathscr{L}(\mathscr{T})$, if considered as sets.

As the measure of complexity of a tree $\mathscr{T} \subset \mathscr{T}^*$ we consider the number of subdivisions $\mathbf{n}(\mathscr{T})$ needed to create $\mathscr{T}$ from its root. We shall often use the fact that

$$\mathbf{n}(\mathscr{T}) = \#(\mathscr{N}(\mathscr{T})). \tag{33}$$

It follows that

$$\#(\mathscr{T}) = K\mathbf{n}(\mathscr{T}) + 1 \tag{34}$$

Also, for a complete tree, $\mathscr{L}(\mathscr{T}) = 1 + (K-1)\mathbf{n}(\mathscr{T})$. So, $\mathbf{n}(\mathscr{T})$ is a fair measure of the complexity of $\mathscr{T}$.

In tree approximation, we assume that to every node $I \in \mathscr{T}^*$, we have an error or *energy* $e(I)$. We measure the performance of a finite tree $\mathscr{T}$ by

$$E(\mathscr{T}) := \sum_{I \in \mathscr{L}(\mathscr{T})} e(I). \tag{35}$$

If we are considering trees corresponding to adaptive partitioning then we would take $e(I) = E(f,I)_p^p$ where $E(f,I)_p$ is the local $L_p(I)$ error on the cell I. Similarly, if we are doing wavelet approximation in L_2 then we would take $e(I) := \sum_{J \subset I} \sum_{e \in E} |f_I^e|^2$ which would be the energy in the wavelet coefficients on all nodes of the tree below I (this corresponds to the error contributed by not including these coefficients). We are interested in the best performance of trees of size $\mathbf{n}(\mathscr{T}) \leq n$ which is given by

$$\sigma_n := \inf_{\mathbf{n}(\mathscr{T}) \leq n} E(\mathscr{T}). \tag{36}$$

Using this definition of σ_n gives the approximation classes $\mathscr{A}_q^r(L_p)$ for tree approximation in L_p.

What is the cost of tree approximation versus n-term approximation? The main point of our work with Wolfgang on wavelet tree approximation given in [20] is

that the cost is almost negligible. Recall that for n-term wavelet approximation in $L_p(\Omega)$, $\Omega \subset \mathbb{R}^d$, we achieve error $O(n^{-r/d})$ for a function f if it is in the Besov space $B^r_q(L_q(\mathbb{R}^d))$ with $1/q = r/d + 1/p$. These latter spaces are barely embedded in L_p and are not compactly embedded. We prove in [20] that whenever a Besov space $B^r_q(L_\tau)$ is compactly embedded into $L_p(\Omega)$ then wavelet tree approximation gives the same approximation rate $O(n^{-r/d})$. Said in another way, this Besov space is embedded into $\mathcal{A}_\infty^{r/d}(L_p)$. Of course, we get such a compact embedding whenever $\tau > (r/d + 1/p)^{-1}$ because of the Sobolev embedding theorem. Thus, from this point of view, tree approximation performs almost as well as n-term approximation.

The proof of the above result on the performance of wavelet tree approximation requires the counting of the new nodes added in order to guarantee the tree structure. However, the number of these new nodes can be controlled by grouping the nodes according to the size of the wavelet coefficients and counting each grouping. Finally, let us remark that in [11] we prove similar theorems on tree approximation for trees generated by adaptive partitioning. This plays an important role in understanding which solutions to elliptic partial differential equations can be well approximated by adaptive finite element methods.

Let us turn to the discussion of finding near best trees. Finding the best tree that matches σ_k in (36) is practically infeasible since it would require searching over all trees $\mathcal{T} \subset \mathcal{T}^*$ with $\mathbf{n}(\mathcal{T}) = k$ and the number of such trees is exponential in k. Remarkably, however, it is possible to design practical algorithms that do almost as well while involving only $O(n)$ computations. The first algorithms of this type were given in [12]. We shall describe a modification of this approach that gives slightly better constants in the estimation of performance.

The tree algorithm we shall consider can be implemented in the general setting of [12]. However, here, we shall limit ourselves to the following setting. We assume the error functionals are *subadditive* in the sense that

$$e(\mathbf{I}) \geq \sum_{\mathbf{I}' \in \mathcal{C}(\mathbf{I})} e(\mathbf{I}'), \tag{37}$$

where $\mathcal{C}(\mathbf{I})$ is the set of children of $\mathbf{I}$. This property holds for the examples we have described above.

A naive strategy to generate a good tree for adaptive approximation would be to mark for subdivision the cells which have largest local errors. However, such a strategy would not generate near optimal trees because it could happen that subdividing a cell and its successive generations would not reduce at all the global error and so a better strategy would have been to subdivide some other cell. To obtain near optimal algorithms, one has to be more clever and penalize successive subdivisions which do not markedly reduce the error. This is done through certain *modified error functionals* $\tilde{e}(\mathbf{I})$ whose precise definition we postpone for a moment. The tree algorithm we propose will grow a given tree $\mathbb{T}$ by including the children of $\mathbf{I}$ as new nodes when $\tilde{e}(\mathbf{I})$ is the largest among all $\tilde{e}(\mathbf{I}') \in \Lambda(\mathbb{T})$.

In our formulation and analysis of the tree algorithm, the local error functional e can be any functional defined on the nodes $\mathbf{I}$ in $\mathcal{T}$ which is subadditive.

Tree-Algorithm:

- Let $\mathscr{T}^0 := \{X\}$ be the root tree.
- If $\mathscr{T}^k$ has been defined for some $k \geq 0$, then define

$$\mathbf{I}^* = \operatorname{argmax}\{\tilde{e}(\mathbf{I}) : \mathbf{I} \in \mathscr{L}(\mathscr{T}_k)\}$$

and $\mathscr{T}^{k+1} := \mathscr{T}^k \cup \{\mathscr{C}(\mathbf{I}^*)\}$.

As the modified error functional, we employ

$$\tilde{e}(\mathbf{I}) := e(\mathbf{I}) \quad \text{for} \quad \mathbf{I} = X \quad \text{and} \quad \tilde{e}(\mathbf{I}) := \left(\frac{1}{e(\mathbf{I})} + \frac{1}{\tilde{e}(\mathbf{I}')}\right)^{-1} \quad \text{for} \quad \mathbf{I} \in \mathscr{C}(\mathbf{I}'). \tag{38}$$

The purpose of the modified error is to penalize children of cells which are chosen for subdivision but the resulting refinement does not significantly decrease the total error. Notice that in such a case the modified error $\tilde{e}$ decreases for the children and therefore makes them less apt to be chosen in later subdivisions.

The following theorem describes the performance of the tree algorithm.

Theorem 3.1. *At each step n of the above* tree algorithm *the output tree $\mathbb{I} = \mathbb{I}_n$ satisfies*

$$E(\mathscr{T}) \leq \left(\frac{n}{n-k}\right) \sigma_k, \tag{39}$$

whenever $k < n$.

The main distinction of the above results from previous ones in [12] is that the constant on the right hand side of (39) is now completely specified and, in particular, does not involve the total number of children of a node. Note that the computational complexity of implementing the tree algorithm with a resulting tree $\mathscr{T}$ depends only on $\mathbf{n}(\mathscr{T})$. Therefore, when applying this algorithm to adaptive partitioning, it is independent of the spatial dimension d. The proof of the above theorem will be given in a forthcoming paper with Peter Binev, Wolfgang, and Phillipp Lamby.

3.4 Greedy algorithms

In application domains, there is a desire to have as much approximation power as possible. This is accomplished by choosing a large dictionary $\mathscr{D}$ to increase approximation power. However, their sheer size can cause a stress on computation. Greedy algorithms are a common approach to keeping computational tasks reasonable when dealing with large dictionaries. They have a long history in statistics and signal processing. A recent survey of the approximation properties of such algorithms is given in [51] where one can find the main results of this subject.

We shall consider only the problem of approximating a function f from a Hilbert space $\mathscr{H}$ by a finite linear combination $\hat{f}$ of elements of a given dictionary $\mathscr{D} =$

$(g)_{g\in\mathscr{D}}$. We have already discussed the case where $\mathscr{D}$ is an orthonormal basis. One of the motivations for utilizing general dictionaries rather than orthonormal systems is that in many applications, such as signal processing or statistical estimation, it is not clear which orthonormal system, if any, is best for representing or approximating f. Thus, dictionaries which are a union of several bases or collections of general waveforms are preferred. Some well known examples are the use of Gabor sytems, curvelets, and wavepackets in signal processing and neural networks in learning theory.

When working with dictionaries $\mathscr{D}$ which are not orthonormal bases, the realization of a best n-term approximation is usually out of reach from a computational point of view since it would require minimizing $\|f-\hat{f}\|$ over all $\hat{f}$ in an infinite or huge number of n dimensional subspaces. *Greedy algorithms* or matching pursuit aim to build "sub-optimal yet good" n-term approximations through a greedy selection of elements g_k, $k=1,2,\cdots$, within the dictionary $\mathscr{D}$, and to do so with a more manageable number of computations.

There exist several versions of these algorithms. The four most commonly used are the *pure greedy*, the *orthogonal greedy*, the *relaxed greedy* and the *stepwise projection* algorithms, which we respectively denote by the acronyms PGA, OGA, RGA and SPA. All four of these algorithms begin by setting $f_0 := 0$. We then define recursively the approximant f_k based on f_{k-1} and its residual $r_{k-1} := f - f_{k-1}$.

In the PGA and the OGA, we select a member of the dictionary as

$$g_k := \operatorname{argmax}_{g\in\mathscr{D}} |\langle r_{k-1}, g\rangle|. \tag{40}$$

The new approximation is then defined as

$$f_k := f_{k-1} + \langle r_{k-1}, g_k\rangle g_k, \tag{41}$$

in the PGA, and as

$$f_k = P_k f, \tag{42}$$

in the OGA, where P_k is the orthogonal projection onto $V_k := \operatorname{Span}\{g_1,\cdots,g_k\}$. It should be noted that when $\mathscr{D}$ is an orthonormal basis both algorithms coincide with the computation of the best k-term approximation.

In the RGA, the new approximation is defined as

$$f_k = \alpha_k f_{k-1} + \beta_k g_k, \tag{43}$$

where (α_k,β_k) are real numbers and g_k is a member of the dictionary. There exist many possibilities for the choice of (α_k,β_k,g_k), the most greedy being to select them according to

$$(\alpha_k,\beta_k,g_k) := \operatorname*{argmin}_{(\alpha,\beta,g)\in\mathbb{R}^2\times\mathscr{D}} \|f-\alpha f_{k-1}-\beta g\|. \tag{44}$$

Other choices specify one or several of these parameters, for example by taking g_k as in (40) or by setting in advance the value of α_k and β_k, see e.g. [38] and [4]. Note that the RGA coincides with the PGA when the parameter α_k is set to 1.

In the SPA, the approximation f_k is defined by (42) as in the OGA, but the choice of g_k is made so as to minimize over all $g \in \mathscr{D}$ the error between f and its orthogonal projection onto $\mathrm{Span}\{g_1, \cdots, g_{k-1}, g\}$.

Note that, from a computational point of view, the OGA and SPA are more expensive to implement since at each step they require the evaluation of the orthogonal projection $P_k f$ (and in the case of SPA a renormalization). Such projection updates are computed preferrably using Gram-Schmidt orthogonalization (e.g. via the QR algorithm) or by solving the normal equations

$$G_k a_k = b_k, \tag{45}$$

where $G_k := (\langle g_i, g_j \rangle)_{i,j=1,\cdots,k}$ is the Gramian matrix, $b_k := (\langle f, g_i \rangle)_{i=1,\cdots,k}$, and $a_k := (\alpha_j)_{j=1,\cdots,k}$ is the vector such that $f_k = \sum_{j=1}^{k} \alpha_j g_j$.

In order to describe the known results concerning the approximation properties of these algorithms, we introduce the class $\mathscr{L}_1 := \mathscr{L}_1(\mathscr{D})$ consisting of those functions f which admit an expansion $f = \sum_{g \in \mathscr{D}} c_g g$ where the coefficient sequence (c_g) is absolutely summable. We define the norm

$$\|f\|_{\mathscr{L}_1} := \inf\{ \sum_{g \in \mathscr{D}} |c_g| \ : \ f = \sum_{g \in \mathscr{D}} c_g g \} \tag{46}$$

for this space. This norm may be thought of as an ℓ_1 norm on the coefficients in representation of the function f by elements of the dictionary; it is emphasized that it is not to be confused with the L_1 norm of f. An alternate and closely related way of defining the $\mathscr{L}_1$ norm is by the infimum of numbers V for which f/V is in the closure of the convex hull of $\mathscr{D} \cup (-\mathscr{D})$. This is known as the "variation" of f as introduced in [3].

In the case where $\mathscr{D}$ is an orthonormal basis, we find that if $f \in \mathscr{L}_1$,

$$\sigma_N(f) = (\sum_{g \notin \Lambda_n(f)} |c_g|^2)^{1/2} \leq (\|f\|_{\mathscr{L}_1} \min_{g \in \Lambda_n(f)} |c_g|)^{1/2} \leq \|f\|_{\mathscr{L}_1} N^{-1/2}, \tag{47}$$

which is contained in (23).

For the PGA, it was proved in [29] that $f \in \mathscr{L}_1$ implies that

$$\|f - f_N\| \lesssim N^{-1/6}. \tag{48}$$

This rate was improved to $N^{-\frac{11}{62}}$ in [40], but on the other hand it was shown [43] that for a particular dictionary there exists $f \in \mathscr{L}_1$ such that

$$\|f - f_N\| \gtrsim N^{-0.27}. \tag{49}$$

When compared with (47), we see that the PGA is far from being optimal.

The RGA, OGA and SPA behave somewhat better: it was proved respectively in [38] for the RGA and SPA, and in [29] for the OGA, that one has

$$\|f - f_N\| \lesssim \|f\|_{\mathscr{L}_1} N^{-1/2}, \tag{50}$$

for all $f \in \mathscr{L}_1$.

For each of these algorithms, it is known that the convergence rate $N^{-1/2}$ cannot in general be improved even for functions which admit a very sparse expansion in the dictionary $\mathscr{D}$ (see [29] for such a result with a function being the sum of two elements of $\mathscr{D}$).

At this point, some remarks are in order regarding the meaning of the condition $f \in \mathscr{L}_1$ for some concrete dictionaries. A commonly made statement is that greedy algorithms break the *curse of dimensionality* in that the rate $N^{-1/2}$ is independent of the dimension d of the variable space for f, and only relies on the assumption that $f \in \mathscr{L}_1$. This is not exactly true since in practice the condition that $f \in \mathscr{L}_1$ becomes more and more stringent as d grows. For instance, in the case where we work in the Hilbert space $\mathscr{H} := L_2([0,1]^d)$ and when $\mathscr{D}$ is a *wavelet basis* (ψ_λ), it follows from our earlier observations in §3.1 that the smoothness property which ensures that $f \in \mathscr{L}_1$ is that f should belong to the Besov space $B_1^s(L_1)$ with $s = d/2$, which roughly means that f has all its derivatives of order less or equal to $d/2$ in L_1 (see [25] for the characterization of Besov spaces by the properties of wavelet coefficients). Another instance is the case where $\mathscr{D}$ consists of sigmoidal functions of the type $\sigma(v \cdot x - w)$ where σ is a fixed function and v and w are arbitrary vectors in $\mathbb{R}^d$, respectively real numbers. For such dictionaries, it was proved in [4] that a sufficient condition to have $f \in \mathscr{L}_1$ is the convergence of $\int |\omega||\mathscr{F} f(\omega)|d\omega$ where $\mathscr{F}$ is the Fourier operator. This integrability condition requires a larger amount of decay on the Fourier transform $\mathscr{F} f$ as d grows. Assuming that $f \in \mathscr{L}_1$ is therefore more and more restrictive as d grows. Similar remarks also hold for other dictionaries (hyperbolic wavelets, Gabor functions etc.).

The above discussion points to a significant weakness in the theory of greedy algorithms in that there are no viable bounds for the performance of greedy algorithms for general functions $f \in \mathscr{H}$. This is a severe impediment in some application domains (such as learning theory) where there is no a priori knowledge that would indicate that the target function is in $\mathscr{L}_1$. One of the main contributions of the work with Wolfgang [7] was to provide error bounds for the performance of greedy algorithms for general functions $f \in \mathscr{H}$. This was accomplished by developing a technique based on interpolation of operators that provides convergence rates N^{-s}, $0 < s < 1/2$, whenever f belongs to a certain intermediate space between $\mathscr{L}_1$ and the Hilbert space $\mathscr{H}$. Namely, we used the spaces

$$\mathscr{B}_p := [\mathscr{H}, \mathscr{L}_1]_{\theta,\infty}, \quad \theta := 2/p - 1, \quad 1 < p < 2, \tag{51}$$

which are the real interpolation spaces between $\mathscr{H}$ and $\mathscr{L}_1$. We showed that if $f \in \mathscr{B}_p$, then the OGA and RGA, when applied to f, provide approximation rates CN^{-s} with $s := \theta/2 = 1/p - 1/2$. Thus, if we set $\mathscr{B}_1 = \mathscr{L}_1$, then these spaces provide a full range of approximation rates for greedy algorithms. Recall, as discussed previously, for general dictionaries, greedy algorithms will not provide convergence rates better than $N^{-1/2}$ for even the simplest of functions. The results we obtained were optimal in the sense that they recovered the best possible convergence rate in

the case where the dictionary is an orthonormal basis. For an arbitrary target function $f \in \mathscr{H}$, convergence of the OGA and RGA holds without rate.

4 Image compression

The emergence of wavelets as a good representation system took place in the late 1980's. One of the most impressive applications of the wavelet system occured in image processing, especially compression and denoising. There are a lot of stories to be told here including the method of thresholding wavelet coefficients for denoising, first suggested by Donoho and Johnstone [31], as a simple methodology for effectively solving imaging problems. But we shall restrict our attention to the problem of understanding the best implementation of wavelets in compression (image encoding).

What is an image? Too often the view is a digitized image. While this matches what we treat in application, it is not the correct launching point for a theory. Engineers usually view images and signals as realizations of a stochastic process. One can debate the efficacy of this viewpoint versus the deterministic viewpoint I am going to now advocate.

In [26], we proposed to view images as functions f defined on a continuum which we shall normalize as the unit square $[0,1]^2$. The digitized images we observe are then simply samples of f given as averages over small squares (pixels). Thus, any representation system for functions on $[0,1]^2$ can be used to for images and computations are made from the samples. We advocated the use of wavelets because of its multiscale structure and the remainder of our discussion of image processing will be limited to wavelet decompositions.

Suppose we we wish to compress functions using wavelet decompositions. The first step is to chose the norm or metric in which we wish to measure distortion. This is traditionally done using the L_2 norm which corresponds to what Engineers use in their measure of Peak Signal to Noise Ratio (PSNR). However, for the purposes of this discussion any L_p norm would work equally well. We have already seen that a near best n term approximation (actually best when $p = 2$) is gotten by simply keeping the n largest terms (measured in L_p) of the wavelet decomposition. So this must be how to do compression. However to convert everything to a binary bitstream one has to further quantize the coefficients since in general the wavelet coefficients are real numbers.

Understanding how to quantize is quite easy if one recalls the connection between n-term approximation and thresholding. Namely, as explained earlier, except for possible ties in the sizes of wavelet coefficients, choosing the biggest n terms corresponds to setting a threshold and retaining the wavelet coefficients above this threshold. Since thresholding takes the view that coefficients below the threshold size $\eta > 0$ should not be retained, it makes perfect sense that quantizing a wavelet coefficient a should be made by taking the smallest number of binary bits of a so that the recovery $\hat{a}$ from these bits satisfies $|a - \hat{a}| \leq \eta$. This makes a perfectly reason-

able compression scheme except that in addition one has to send bits to identify the index of the wavelet coefficient. Here the matter becomes a little more interesting.

Before embarking on the index identification problem, let us remark that the characterization (given in §3.1) of the approximation classes $\mathscr{A}_\tau^r(L_p)$ as Besov spaces $B_\tau^s(L_\tau)$ when $1/\tau = r + 1/p$ and $r = s/2$ (because we are in two space dimensions) gives a very satisfying characterization of which images can be compressed with a given distortion rate if we measure complexity of the encoding by the number of terms retained in the wavelet decomposition. This was the story told in [26]. However, there was rightfully considerable objection to this theory since it was based on the number of terms n retained and not on the number of bits needed to encode this information.

A major step in the direction of giving a theory based on the number of bits was taken in the paper of Cohen, Daubechies, Gulyeruz, and Orchard [21]. It was however limited to measuring distortion in the L_2 norm. With Wolfgang, we wanted to give a complete theory that would include measuring distortion in any L_p space. The key step in developing such a theory was to consider the notion of tree approximation and in fact this is where the theory of tree approximation characterizing the spaces $\mathscr{A}_q^r(L_p, tree)$ for the wavelet basis (described earlier) was developed. Let us see how this solves our encoding problem.

To build a compression for functions, we first choose our compression metric L_p. We then agree on a minimal smoothness ε that we shall assume of the functions in L_p. This step is necessary so that the encoder is applied to a compact set of functions. Next, we find the wavelet coefficients of the wavelet decomposition of the image with respect to the wavelet basis normalized in L_p. We then build a sequence of trees $\mathscr{T}_k$ associated to the image as follows. We consider the set Λ_k of all wavelet indices for which the coefficient of the image is in absolute value $\geq 2^{-k}$. The nodes in Λ_k will not form a tree so we complete them to the smallest tree $\mathscr{T}_k$ which contains Λ_k. An important point here is that the sets Λ_k and the tree $\mathscr{T}_k$ can be found without computing and searching over an infinite set of wavelet coefficients because of our assumption on minimal smoothness in L_p.

Notice that the tree $\mathscr{T}_k$ is contained in $\mathscr{T}_{k+1}$. Therefore $\Delta_k := \mathscr{T}_k \setminus \mathscr{T}_{k-1}$ will tell us how to obtain $\mathscr{T}_k$ once $\mathscr{T}_{k-1}$ is known. This process is called *growing the tree*.

We shall send a progressive bitstream to the receiver. After receiving any portion of this bitstream the receiver will be able to construct an approximation of the image with higher and higher resolution (in our chosen L_p metric) as more and more bits are received. The first bits will identify the smallest value of k_0 for which Λ_{k_0} is nonempty. Then come the bits to identify $\mathscr{T}_{k_0}$ followed by bits to identify the sign of the coefficients in $\mathscr{T}_{k_0}$ and one bit of the binary expansion of each of the coefficients. Later bits come in packets. Each packet tells us how to go from $\mathscr{T}_{k-1}$ to $\mathscr{T}_k$ and how to increase the resolution of each of the coefficients in hand.

Precisely, in the k-th packet we first send bits that tell how to grow $\mathscr{T}_{k-1}$ to $\mathscr{T}_k$. Next, we send a bit for each new coefficient (i.e. those in Δ_k) to identify its sign, next comes one bit (the lead bit) of the binary expansion for each new coefficient. Finally, we send one additional bit for each of the old coefficients that had been previously sent.

For the resulting encoder one can prove the following result of [20]:
Performance of image encoder: *If the image $f \in B^s_q(L_\tau)$ for some $s > 0$ and $\tau > (s/2+1/p)^{-1}$, then after receiving n bits, these bits can be decoded to give an image $\hat{f}$ such that $\|f - \hat{f}\|_{L_p} \leq Cn^{-s/2}$.*

There were two key ingredients in proving the above result on the performance of the encoder. The first of these is to show that tree approximation is as effective as n-term approximation when approximating functions in Besov classes that compactly embed into L_p. We have already discussed this issue in our section on tree approximation. The second new ingredient is to show that any quad tree with m nodes can be encoded using at most $4m$ bits. Here, we borrowed the ideas from [21].

5 Remarks on nonlinear approximation in PDE solvers

Certainly, the construction of numerical algorithms based on nonlinear approximation for solving PDEs has been one of Wolfgang's major accomplishments. An extensive description of this development for elliptic PDEs will be presented in the contribution of Morin, Nochetto and Siebert in this volume. We will restrict our remarks to some historical comments.

We shall discuss only the model Laplace problem

$$-\Delta(u) = f \text{ on } \Omega, \quad u = 0 \text{ on } \partial\Omega, \tag{52}$$

where $f \in H^{-1}$ and the solution u is to be captured in the energy norm which in this case is the $H^1_0(\Omega)$ norm. The solution to such equations is well known to generate singularities of two types. The first is due to singularities in f itself while the other come from the boundary of the domain, for example corner singularities. So it is natural to envision nonlinear approximation methods as the basis for effective numerical solvers. Indeed, it was already shown in [23], that the solutions to (52) on Lipschitz domains always have higher smoothness in the scale of Besov spaces corresponding to nonlinear approximation than they do in the scale for linear approximation. So the theoretical underpinnings were there to advocate nonlinear methods and they were certainly in vogue beginning with the work of Ivo Babuska and his collaborators (starting with [1]). Surprisingly, there was no algorithm based on nonlinear methods which was proven to outperform linear methods save for some univariate results.

Wolfgang brought Albert and I this problem and explained the bulk chasing technique of Doerfler [32] which can be used to show convergence (but no rates) for adaptive finite element methods (with some massaging as provided by Morin, Nochetto, and Siebert [44]). We thought that the easiest type algorithm to analyze would be based on wavelet decompositions. One advantage of chosing wavelets is that (52) can be converted to an infinite matrix operator equation

$$\mathcal{A}\bar{u} = \bar{f} \tag{53}$$

where $\mathscr{A}$ is bounded and boundedly invertible on ℓ_2. Here one employs the wavelet preconditioning (diagonal rescaling) utilized in the analysis of preconditioning in [24]. The key property inherited by this matrix is off diagonal decay which can also be described as a compressibility in that $\mathscr{A}$ can be well approximated by finite rank matrices.

In analogy with the results on image encoding, we wanted to create a Galerkin algorithm for numerically solving (52) based on wavelet tree approximation such that whenever u is in one of the approximation classes $\mathscr{A}^s$ then the algorithm produces an approximant to u (in the energy norm) with near optimal rate distortion. Namely, if N is the cardinality of the tree $\mathscr{T}$ associated to the numerical approximation $u_{\mathscr{T}}$, then

$$\|u - u_{\mathscr{T}}\|_{H^1_0} \leq C_0 \|u\|_{\mathscr{A}^s} N^{-s}. \tag{54}$$

In the end we actually did much better since we showed the operational count needed to compute $u_{\mathscr{T}}$ could also be kept proportional to N.

We were quickly able to build the framework for the wavelet numerical algorithm. However, we wrestled for quite some time to derive optimal bounds for the number of terms in the wavelet decomposition of the approximant. This of course is necessary for any rate distortion theory. In the end, we went back to our analogy with image compression where one discards small coefficients in such decompositions when seeking optimal compression and noise reduction. This led to our coarsening algorithm and a subsequent proof of optimal performance of the numerical algorithm. It was an important contribution of Stevenson [48] that it is actually possible to build adaptively wavelet algorithms without coarsening with the same optimal rate distortion theory. Heuristically, if one is not too aggressive with the bulk chasing then the majority of the nodes chosen will in the end survive coarsening.

Our first paper [16] on adaptive wavelet methods was built on solving finite discrete problems formed by taking appropriate subsections of the matrix $\mathscr{A}$. This actually turned out to be the wrong view. Wolfgang proposed the idea that we should retain as long as possible the infinite matrix form (53) and algorithms should be viewed as solving this infinite dimensional problem. This turned out to be not only the right conceptual view but also very powerful in algorithm development. This allowed us to solve non-coercive problems and provide a very robust and elegant theory in [17].

With Peter Binev, Wolfgang and I wondered why we could not carry our wavelet theory over to finite element methods based on adaptive triangulations . We quickly found out that these algorithms had major differences from wavelet algorithms. First of all, in contrast to having one matrix (53) governing the algorithm, the matrices changed at each iteration. This made the effect of refining triangles much more subtle than the growing wavelet trees. Fortunately, we were able to borrow the theory of local error estimators for finite elements developed by Morin, Nochetto, and Siebert [44]. Another major difficulty was the fact the problem of hanging nodes (or non-conforming elements). This required us to develop a way to count the additional refinements necessary to guarantee conforming elements. This was eventually given

by a nice maximal function type algorithm. Our algorithm for adaptive finite element methods again had a coarsening step based on the tree algorithm of [12]. Again, Rob Stevenson was able to show that one can proceed without coarsening. Now there is a much finer understanding of adaptive finite element algorithms which will be well presented in the contribution of Morin, Nochetto, and Siebert in this volume.

6 Learning theory

Learning theory is a problem in data fitting. The data is assumed to be generated by an unknown measure ρ defined on a product space $Z := X \times Y$. We shall assume that X is a bounded domain of $\mathbb{R}^d$ and $Y = \mathbb{R}$. The article of Gerard Kerkyacharian, Mathilde Mougeot, Dominique Picard, and Karine Tribouley in this volume will give a general exposition of this subject. Here we want to touch on some aspects of this subject that relate to nonlinear approximation.

We assume that we are given m independent random observations $z_i = (x_i, y_i)$, $i = 1, \dots, m$, identically distributed according to ρ. We are interested in finding the function f_ρ which best describes the relation between the y_i and the x_i. This is the *regression function* $f_\rho(x)$ defined as the conditional expectation of the random variable y at x:

$$f_\rho(x) := \int_Y y d\rho(y|x) \tag{55}$$

with $\rho(y|x)$ the conditional probability measure on Y with respect to x. We shall use $\mathbf{z} = \{z_1, \dots, z_m\} \subset Z^m$ to denote the set of observations.

One of the goals of learning is to provide estimates under minimal restrictions on the measure ρ since this measure is unknown to us. We shall work under the mild assumption that this probability measure is supported on an interval $[-M, M]$

$$|y| \le M, \tag{56}$$

almost surely. It follows in particular that $|f_\rho| \le M$. This property of ρ can usually be inferred in practical applications.

We denote by ρ_X the marginal probability measure on X defined by

$$\rho_X(S) := \rho(S \times Y). \tag{57}$$

We shall assume that ρ_X is a Borel measure on X. We have

$$d\rho(x,y) = d\rho(y|x) d\rho_X(x). \tag{58}$$

It is easy to check that f_ρ is the minimizer of the risk functional

$$\mathscr{E}(f) := \int_Z (y - f(x))^2 d\rho, \tag{59}$$

over $f \in L_2(X,\rho_X)$ where this space consists of all functions from X to Y which are square integrable with respect to ρ_X. In fact one has

$$\mathscr{E}(f) = \mathscr{E}(f_\rho) + \|f - f_\rho\|^2, \tag{60}$$

where

$$\|\cdot\| := \|\cdot\|_{L_2(X,\rho_X)}. \tag{61}$$

The goal in learning is to find an *estimator* $f_{\mathbf{z}}$ for f_ρ from the given data $\mathbf{z}$. The usual way of evaluating the performance of such an estimator is by studying its convergence either in probability or in expectation, i.e. the rate of decay of the quantities

$$\text{Prob}\{\|f_\rho - f_{\mathbf{z}}\| \geq \eta\}, \quad \eta > 0 \quad \text{or} \quad E(\|f_\rho - f_{\mathbf{z}}\|^2) \tag{62}$$

as the sample size m increases. Here both the expectation and the probability are taken with respect to the product measure ρ^m defined on Z^m. Estimations in probability are to be preferred since they give more information about the success of a particular algorithm and they automatically yield an estimate in expectation by integrating with respect to η. Much more is known about the performance of algorithms in expectation. This type of regression problem is referred to as *random design* or *distribution-free* because there are no a priori assumption on ρ_X. An excellent survey on distribution free regression theory is provided in the book [35], which includes most existing approaches as well as the analysis of their rate of convergence in the expectation sense.

A common approach to regression estimation is to choose an hypothesis (or *model*) class $\mathscr{H}$ and then to define $f_{\mathbf{z}}$, in analogy to (59), as the minimizer of the empirical risk

$$f_{\mathbf{z}} := \operatorname*{argmin}_{f\in\mathscr{H}} \mathscr{E}_{\mathbf{z}}(f), \quad \text{with} \quad \mathscr{E}_{\mathbf{z}}(f) := \frac{1}{m}\sum_{j=1}^{m}(y_j - f(x_j))^2. \tag{63}$$

In other words, $f_{\mathbf{z}}$ is the best approximation to $(y_j)_{j=1}^m$ from $\mathscr{H}$ in the the empirical norm

$$\|g\|_m^2 := \frac{1}{m}\sum_{j=1}^{m}|g(x_j)|^2. \tag{64}$$

Typically, $\mathscr{H} = \mathscr{H}_m$ depends on a finite number $n = n(m)$ of parameters. Of course, we advocate the use of nonlinear families $\mathscr{H}_m$ for the reasons already made abundantly clear in this exposition. In some algorithms, the number n is chosen using an a priori assumption on f_ρ. Better algorithms avoid such prior assumptions and the number n is adapted to the data in the algorithm. This is usually done by what is called model selection in statistics but this can be sometimes be an expensive numerical procedure in practical implementations.

Estimates for the decay of the quantities in (62) are usually obtained under certain assumptions (called *priors*) on f_ρ. We emphasize that the algorithms should not depend on prior assumptions on f_ρ. Only in the analysis of the algorithms do we impose such prior assumptions in order to see how well the algorithm performs.

Priors on f_ρ are typically expressed by a condition of the type $f_\rho \in \Theta$ where Θ is a class of functions that necessarily must be contained in $L_2(X, \rho_X)$. If we wish the error, as measured in (62), to tend to zero as the number m of samples tends to infinity then we necessarily need that Θ is a compact subset of $L_2(X, \rho_X)$. There are three common ways to measure the compactness of a set Θ: (i) minimal coverings, (ii) smoothness conditions on the elements of Θ, (iii) the rate of approximation of the elements of Θ by a specific approximation process.

In studying the estimation of the regression function, the question arises at the outset as to what are the best approximation methods to use in deriving algorithms for approximating f_ρ and therefore indirectly in defining prior classes? With no additional knowledge of ρ (and thereby f_ρ) there is no general answer to this question. This is in contrast to numerical methods for PDEs where regularity theorems for the PDEs can lead to the optimal recovery schemes.

However, it is still possible in learning to draw some distinctions between certain strategies. Suppose that we seek to approximate f_ρ by the elements from a hypothesis class $\mathscr{H} = \Sigma_n$. Here the parameter n measures the complexity associated to the process. In the case of approximation by elements from linear spaces we will take the space Σ_n to be of dimension n. For nonlinear methods, the space Σ_n is not linear and now n represents the number of parameters used in the approximation.

If we have two approximation methods corresponding to sequences of approximation spaces (Σ_n) and (Σ_n'), then the second process would be superior to the first in terms of rates of approximation if $E_n'(g) \le CE_n(g)$ for all g and an absolute constant $C > 0$. For example, approximation using piecewise linear functions would in this sense be superior to using approximation by piecewise constants. In our learning context however, there are other considerations since: (i) the rate of approximation need not translate directly into results about estimating f_ρ because of the uncertainty in our observations, (ii) it may be that the superior approximation method is in fact much more difficult (or impossible) to implement in practice. For example, a typical nonlinear method may consist of finding an approximation to g from a family of linear spaces each of dimension N. The larger the family the more powerful the approximation method. However, too large of a family will generally make the numerical implementation of this method of approximation impossible.

Suppose that we have chosen the space Σ_n to be used as our hypothesis class $\mathscr{H}$ in the approximation of f_ρ from our given data $\mathbf{z}$. How should we define our approximation? As we have already noted, the most common approach is empirical risk minimization which gives the function $\hat{f}_\mathbf{z} := \hat{f}_{\mathbf{z},\Sigma_n}$ defined by (63). However, since we know $|f_\rho| \le M$, the approximation will be improved if we post-truncate $\hat{f}_\mathbf{z}$ by M. For this, we define the truncation operator

$$T_M(x) := \min(|x|, M)\operatorname{sign}(x) \tag{65}$$

for any real number x and define

$$f_\mathbf{z} := f_{\mathbf{z},\mathscr{H}} := T_M(\hat{f}_{\mathbf{z},\mathscr{H}}). \tag{66}$$

There are general results that provide estimates for how well $f_{\mathbf{z}}$ approximates f_ρ. One such estimate given in [35] (see Theorem 11.3) applies when $\mathscr{H}$ is a linear space of dimension n and gives

$$E(\|f_\rho - f_{\mathbf{z}}\|^2) \lesssim \frac{n\log(m)}{m} + \inf_{g\in\mathscr{H}} \|f_\rho - g\|^2. \tag{67}$$

The second term is the bias and equals our approximation error $E_n(f_\rho)$ for approximation using the elements of $\mathscr{H}$. The first term is the variance which bounds the error due to uncertainty. One can derive rates of convergence in expectation by balancing both terms (see [35] and [27]) for specific applications.

The deficiency of this approach is that one needs to know the behavior of $E_n(f_\rho)$ in order to choose the best value of n and this requires a priori knowledge of f_ρ. There is a general procedure known as model selection which circumvents this difficulty and tries to automatically choose a good value of n (depending on f_ρ) by introducing a penalty term. Suppose that $(\Sigma_n)_{n=1}^m$ is a family on linear spaces each of dimension n. For each $n = 1,2,\dots,m$, we have the corresponding function $f_{\mathbf{z},\Sigma_n}$ defined by (66) and the empirical error

$$\hat{E}_{n,\mathbf{z}} := \frac{1}{m}\sum_{j=1}^m (y_j - f_{\mathbf{z},\Sigma_n}(x_j))^2. \tag{68}$$

Notice that $E_{n,\mathbf{z}}$ is a computable quantity which we can view as an estimate for $E_n(f_\rho)$. In complexity regularization, one chooses a value of n by

$$n^* := n^*(\mathbf{z}) := \operatorname{argmin}\{E_{n,\mathbf{z}} + \frac{n\log m}{m}\}. \tag{69}$$

We now define

$$\hat{f}_{\mathbf{z}} := f_{\mathbf{z},\Sigma_{n^*}} \tag{70}$$

as our estimator to f_ρ. One can then prove (see Chapter 12 of [35]) that whenever f_ρ can be approximated to accuracy $E_n(f_\rho) \le Mn^{-s}$ for some $s > 0$, then

$$E(\|f_\rho - f_{\mathbf{z}}\|^2_{L_2(X,\rho_X)}) \le C[\frac{(\log m)^2}{m}]^{\frac{2s}{2s+1}} \tag{71}$$

which save for the logarithm is an optimal rate estimation in expectation. For a certain range of s, one can also prove similar estimates in probability (see [27]). Notice that the estimator did not need to have knowledge of s and nevertheless obtains the optimal performance.

Model selection can also be applied in the setting of nonlinear approximation, i.e. when the spaces Σ_n are nonlinear but in this case, one needs to invoke conditions on the compatibility of the penalty with the complexity of the approximation process as measured by an entropy restriction. We refer the reader to Chapter 12 of [35] for a more detailed discussion of this topic

Let us also note that the penalty approach is not always compatible with the practical requirement of *on-line* computations. By on-line computation, we mean that the estimator for the sample size m can be derived by a simple update of the estimator for the sample size $m-1$. In penalty methods, the optimization problem needs to be globally re-solved when adding a new sample. However, when there is additional structure in the approximation process such as the adaptive partitioning, then there are algorithms that circumvent this difficulty.

With Wolfgang, we wanted to develop algorithms based on nonlinear piecewise polynomials which are universally optimal and in addition are numerically easy to implement. Our first paper [9] built such an algorithm based on piecewise constant approximation. Its implementation is very simple (wavelet like) and can be done on line with streaming data. We proved theorems which showed the optimality of this algorithm in terms of the desirable probability estimates.

While proving the results in [9], we were puzzled by the fact that these results did not carry over nontrivially to general piecewise polynomials. Through a family of counterexamples, we found that if we wanted estimators which perform well in probability then either we must assume something more about the underlying probability measure ρ or we must find an alternative to empirical risk minimization. The simplest way out of this dilemma was to use post truncation as described in (66). Using this type of truncation, we developed in [7] optimal adaptive partitioning learning algorithms for arbitrary polynomial degrees and proved their universal optimality.

6.1 Learning with greedy algorithms

We have already emphasized that keeping the computational task reasonable in learning algorithms is a significant issue. For this reason, with Wolfgang we studied the application of greedy algorithms for learning. The main goal of our extension of the theory of greedy algorithms, as discussed in §3.4 was to apply these to the learning problem. Indeed, we built an estimator based on the application of the OGA or RGA to the noisy data (y_i) in the Hilbert space defined by the empirical norm

$$\|f\|_n := \frac{1}{n}\sum_{i=1}^{n} |f(x_i)|^2, \tag{72}$$

and its associated inner product. At each step k, the algorithm generates an approximation $\hat{f}_k$ to the data. Our estimator was then defined by

$$\hat{f} := T\hat{f}_{k^*} \tag{73}$$

where T is the truncation operator (65) and the value of k^* is selected by a complexity regularization procedure. Our main result for this estimator was (roughly) that when the regression function f_ρ is in $\mathscr{B}_p$ (where this space is defined with respect

to the norm $\|u\|^2 := E(|u(x)|^2)$ as in §3.4), the estimator has convergence rate

$$E(\|\hat{f} - f_\rho\|^2) \lesssim \left(\frac{n}{\log n}\right)^{-\frac{2s}{1+2s}}, \tag{74}$$

again with $s := 1/p - 1/2$. In the case where $f_\rho \in \mathscr{L}_1$, we obtain the same result with $p = 1$ and $s = 1/2$. We also show that this estimator is universally consistent.

In order to place these results into the current state of the art of statistical learning theory, let us first remark that similar convergence rate for the denoising and the learning problem could be obtained by a more "brute force" approach which would consist in selecting a proper subset of $\mathscr{D}$ by complexity regularization with techniques such as those in [2] or Chapter 12 of [35]. Following for instance the general approach of [35], this would typically first require restricting the size of the dictionary $\mathscr{D}$ (usually to be of size $O(n^a)$ for some $a > 1$) and then considering all possible subsets $\Lambda \subset \mathscr{D}$ and spaces $\mathscr{G}_\Lambda := \mathrm{Span}\{g \in \Lambda\}$, each of them defining an estimator

$$\hat{f}_\Lambda := T\left(\mathrm{Argmin}_{f \in \mathscr{G}_\Lambda} \|y - f\|_n^2\right) \tag{75}$$

The estimator $\hat{f}$ is then defined as the $\hat{f}_\Lambda$ which minimizes

$$\min_{\Lambda \subset \mathscr{D}} \{\|y - \hat{f}_\Lambda\|_n^2 + \mathrm{Pen}(\Lambda, n)\} \tag{76}$$

with $\mathrm{Pen}(\Lambda, n)$ a complexity penalty term. The penalty term usually restricts the size of Λ to be at most $\mathscr{O}(n)$ but even then the search is over $O(n^{an})$ subsets. In some other approaches, the sets $\mathscr{G}_\Lambda$ might also be discretized, transforming the subproblem of selecting $\hat{f}_\Lambda$ into a discrete optimization problem.

The main advantage of using the greedy algorithm in place of (76) for constructing the estimator is a dramatic reduction of the computational cost. Indeed, instead of considering all possible subsets $\Lambda \subset \mathscr{D}$ the algorithm only considers the sets $\Lambda_k := \{g_1, \cdots, g_k\}$, $k = 1, \ldots, n$, generated by the empirical greedy algorithm. This approach was proposed and analyzed in [41] using a version of the RGA in which

$$\alpha_k + \beta_k = 1 \tag{77}$$

which implies that the approximation f_k at each iteration stays in the convex hull $\mathscr{C}_1$ of $\mathscr{D}$. The authors established that if f does not belong to $\mathscr{C}_1$, the RGA converges to its projection onto $\mathscr{C}_1$, In turn, the estimator was proved to converge in the sense of (74) to f_ρ, with rate $(n/\log n)^{-1/2}$, if f_ρ lies in $\mathscr{C}_1$, and otherwise to its projection onto $\mathscr{C}_1$. In that sense, this procedure is not universally consistent.

Our main contribution in the work with Wolfgang was to remove requirements of the type $f_\rho \in \mathscr{L}_1$ when obtaining convergence rates. In the learning context, there is indeed typically no advanced information that would guarantee such restrictions on f_ρ. The estimators that we construct for learning are now universally consistent and have provable convergence rates for more general regression functions described by means of interpolation spaces. One of the main ingredient in our analysis of the performance of our greedy algorithms in learning is a powerful exponential con-

centration inequality which was introduced in [41]. Let us mention that a closely related analysis, which however does not involve interpolation spaces, was developed in [5, 6].

Let us finally mention that there exist some natural connections between the greedy algorithms which we have discussed and other numerical techniques for building a sparse approximation in the dictionnary based on the minimization of an ℓ_1 criterion. In the statistical context, these are the celebrated LASSO [52, 36] and LARS [33] algorithms. The relation between ℓ_1 minimization and greedy selection is particularly transparent in the context of deterministic approximation of a function f in an orthonormal basis: if we consider the problem of minimizing

$$\|f - \sum_{g\in\mathscr{D}} d_g g\|^2 + t \sum_{g\in\mathscr{D}} |d_g| \tag{78}$$

over all choices of sequences (d_g), we see that it amounts in minimizing $|c_g - d_g|^2 + t|d_g|$ for each individual g, where $c_g := \langle f, g\rangle$. The solution to this problem is given by the *soft thresholding* operator

$$d_g := c_g - \frac{t}{2}\mathrm{sign}(c_g) \text{ if } |c_g| > \frac{t}{2}, \ 0 \text{ else}, \tag{79}$$

and is therefore very similar to picking the largest coefficients of f.

7 Compressed sensing

Compressed sensing came into vogue during the last few years but its origins lie in results from approximation and functional analysis dating back to the 1970's. The primary early developers were Kashin [39] and Gluskin [34]. Donoho [30] and Candés and Tao [14] showed the importance of this theory in signal processing and added substantially to the theory and its numerical implementation, especially how to do decoding in a practical way.

In discrete compressed sensing, we want to capture a vector (signal) $x \in \mathbb{R}^N$ with N large. Of course if we make N measurements we will know x exactly. The problem is to make comparably fewer measurements and still have enough information to accurately recover x. Since the subject is intimately intertwined with sparsity and nonlinear approximation, the problems of compressed sensing immediately peaked our interest.

The m measurements we are allowed to make about x are of the form of an inner product of x with prescribed vectors. These measurements are represented by a vector

$$y = \Phi x, \tag{80}$$

of dimension $m < N$, where Φ is an $m \times N$ measurement matrix (called a CS matrix). To extract the information that the measurement vector y holds about x, one uses a decoder Δ which is a mapping from $\mathbb{R}^m$ into $\mathbb{R}^N$. The vector $x^* := \Delta(y) = \Delta(\Phi x)$

is our approximation to x extracted from the information y. In contrast to Φ, the operator Δ is allowed to be non-linear.

In recent years, considerable progress has been made in understanding the performance of various choices of the measurement matrices Φ and decoders Δ. Although not exclusively, by far most contributions focus on the ability of such an encoder-decoder pair (Φ,Δ) to recover a *sparse* signal. For example, a typical theorem says that there are pairs (Φ,Δ) such that whenever $x \in \Sigma_k$, with $k \le am/\log(N/k)$, then $x^* = x$.

Our view was that from both a theoretical and a practical perspective, it is highly desirable to have pairs (Φ,Δ) that are robust in the sense that they are effective even when the vector x is not assumed to be sparse. The question arises as to how we should measure the effectiveness of such an encoder-decoder pair (Φ,Δ) for non-sparse vectors. In [18] we have proposed to measure such performance in a metric $\|\cdot\|_X$ by the largest value of k for which

$$\|x-\Delta(\Phi x)\|_X \le C_0\sigma_k(x)_X, \quad \forall x \in I\!R^N, \tag{81}$$

with C_0 a constant independent of k,n,N. We say that a pair (Φ,Δ) which satisfies property (81) is *instance-optimal* of order k with constant C_0. It was shown that this measure of performance heavily depends on the norm employed to measure error. Let us illustrate this by two contrasting results from [18]:

(i) If $\|\cdot\|_X$ is the ℓ_1-norm, it is possible to build encoding-decoding pairs (Φ,Δ) which are instance-optimal of order k with a suitable constant C_0 whenever $m \ge ck\log(N/k)$ provided c and C_0 are sufficiently large. Moreover, the decoder Δ can be taken as

$$\Delta(y) := \operatorname*{argmin}_{\Phi z=y} \|z\|_{\ell_1}. \tag{82}$$

Therefore, in order to obtain the accuracy of k-term approximation, the number m of non-adaptive measurements need only exceed the amount k of adaptive measurements by the small factor $c\log(N/k)$. We shall speak of the range of k which satisfy $k \le am/\log(N/k)$ as the *large range* since it is the largest range of k for which instance-optimality can hold.

(ii) In the case $\|\cdot\|_X$ is the ℓ_2-norm, if (Φ,Δ) is any encoding-decoding pair which is instance-optimal of order $k=1$ with a fixed constant C_0, then the number of measurement m is always larger than aN, where $a>0$ depends only on C_0. Therefore, the number of non-adaptive measurements has to be very large in order to compete with even one single adaptive measurement.

The matrices Φ which have the largest range of instance-optimality for ℓ_1 are all given by stochastic constructions. Namely, one creates an appropriate random family $\Phi(\omega)$ of $m\times N$ matrices on a probability space (Ω,ρ) and then shows that with high probability on the draw, the resulting matrix $\Phi=\Phi(\omega)$ will satisfy instance-optimality for the large range of k. There are no known deterministic constructions. The situation is even worse in the sense that given an $m\times N$ matrix Φ there is no simple method for checking its range of instance-optimality.

While the above results show that instance-optimality is not a viable concept in ℓ_2, it turns out that the situation is not as bleak as it seems. For example, a more optimistic result was established by Candes, Romberg and Tao in [15]. They show that if $m \geq ck\log(N/k)$, it is possible to build pairs (Φ, Δ) such that for all $x \in \mathbb{R}^N$,

$$\|x - \Delta(\Phi x)\|_{\ell_2} \leq C_0 \frac{\sigma_k(x)_{\ell_1}}{\sqrt{k}}, \tag{83}$$

with the decoder again defined by (82). This implies, in particular, that k-sparse signals are exactly reconstructed and that signals x in the space weak ℓ_p with $\|x\|_{w\ell_p} \leq M$ for some $p < 1$ are reconstructed with accuracy $C_0 M k^{-s}$ with $s = 1/p - 1/2$. This bound is of the same order as the best estimate available on $\max\{\sigma_k(x)_{\ell_2} : \|x\|_{w\ell_p} \leq M\}$. Of course, this result still falls short of instance-optimality in ℓ_2 as it must.

What intrigued us was that instance-optimality can be attained in ℓ_2 if one accepts a probabilistic statement. A first result in this direction, obtained by Cormode and Mutukrishnan in [22], shows how to construct random $m \times N$ matrices $\Phi(\omega)$ and a decoder $\Delta = \Delta(\omega)$, $\omega \in \Omega$, such that for any $x \in \mathbb{R}^N$,

$$\|x - \Delta(\Phi x)\|_{\ell_2} \leq C_0 \sigma_k(x)_{\ell_2} \tag{84}$$

holds with overwhelming probability (larger than $1 - \varepsilon(m)$ where $\varepsilon(m)$ tends rapidly to 0 as $m \to +\infty$) as long as $k \leq am/(\log N)^{5/2}$ with a suitably small. Note that this result says that given x, the set of $\omega \in \Omega$ for which (84) fails to hold has small measure. This set of failure will depend on x.

From our viewpoint, *instance-optimality in probability* is the proper formulation in ℓ_2. Indeed, even in the more favorable setting of ℓ_1, we can never put our hands on matrices Φ which have the large range of instance-optimality. We only know with high probability on the draw, in certain random constructions, that we can attain instance-optimality. So the situation in ℓ_2 is not that much different from that in ℓ_1.

The results in [18] pertaining to instance-optimality in probability asked two fundamental questions: (i) can we attain instance-optimality for the largest range of k, i.e. $k \leq an/\log(N/k)$, and (ii) what are the properties of random families that are needed to attain this performance. We showed that instance-optimality can be obtained in the probabilistic setting for the largest range of k, i.e. $k \leq an/\log(N/k)$ using quite general constructions of random matrices. Namely, we introduced two properties for a random matrix Φ which ensure instance-optimality in the above sense and then showed that these two properties hold for rather general constructions of random matrices (such as Gaussian and Bernoulli). However, one shortcoming of the results in [18] is that the decoder used in establishing instance-optimality was defined by minimizing $\|y - \Phi x\|_{\ell_2}$ over all k-sparse vectors, a task which cannot be achieved in any reasonable computational time.

This led us to consider other possible decoders which are numerically friendly and can be coupled with standard constructions of random matrices to obtain an encoding/decoding pair which is instance-optimal for the largest range of k. There are two natural classes of decoders.

The first is based on ℓ_1 minimization as described in (82). It was a nontrivial argument given by Przemek Wojtaczszyk [54] that this decoder gives ℓ_2 instance optimality in probability when coupled with random Gaussian matrices. The key feature of his proof was the fact that such an $m \times N$ Gaussian matrix maps the unit ball in ℓ_1^N onto a set that contains the ball of radius $\frac{\log(N/m)}{m}$ in ℓ_2^m.

The above mapping property fails to hold for general random matrices. For example for the Bernouli family, any point that maps into the vector $e_1 = (1,0,\dots,0)$ must have ℓ_1^N norm $\geq \sqrt{n}$. So some new ideas were needed to prove instance optimality in probability for general random families. This is provided by new mapping properties which state that the image of the unit ℓ_1^N ball covers a certain clipped ℓ_2^N ball. These remarkable mapping properties were first proved in [42] and rediscovered in [28] where the instance optimality is proved.

The other natural decoders for compressed sensing are greedy algorithms. The idea to apply greedy algorithms for compressed sensing originated with Gilbert and Tropp [53] who proposed to use the orthogonal greedy algorithm or orthogonal matching pursuit (OMP) in order to decode y. Namely, the greedy algorithm is applied to the dictionary of column vectors of Φ and the input vector y. After k iterations, it identifies a set of Λ of k column indices (those corresponding to the vectors used to approximate y by the greedy algorithm. Once the set Λ is found, we decode y by taking the minimizer of $\|y - \Phi(z)\|_{\ell_2}$ among all z supported on Λ. The latter step is least squares fitting of the residual and is very fast.

These authors proved the following result for a probabilistic setting for general random matrices which include the Bernouli and Gaussian families: if $m \geq ck\log N$ with c sufficiently large, then for any k sparse vector x, the OMP algorithm returns exactly $x^k = x$ after k iterations, with probability greater than $1 - N^{-b}$ where b can be made arbitrarily large by taking c large enough.

Decoders like OMP are of high interest because of their efficiency. The above result of Gilbert and Tropp remains as the only general statement about OMP in the probabilistic setting. A significant breakthrough on decoding using greedy pursuit was given in the paper of Needel and Vershynin [46] (see also their followup [47]) where they showed the advantage of adjoining a batch of coordinates at each iteration rather than just one coordinate as in OMP. They show that such algorithms can deterministically capture sparse vectors for a slightly smaller range than the largest range of k.

With Wolfgang, we were interested in whether decoders based on thresholding could be used as decoders to yield ℓ_2 instance-optimality in probability for general families of random matrices for the large range of k. In [19] we give an algorithm which does exactly that. This algorithm adds a batch of coordinates at each iteration and then uses a thinning procedure to possibly remove some of them at later iterations. Conceptually, one thinks in terms of a bucket holding all of the coordinates to be used in the construction of x. In the analysis of such algorithms it is important to not allow more than a multiple of k coordinates to gather in the bucket. The thinning is used for this purpose. Thinning is much like the coarsening used in PDE solvers which we described earlier Our algorithm is similar in nature to the COSAMP algorithm of Needel and Tropp [45].

8 Final thoughts

As has been made abundantly clear in this brief survery, Wolfgang Dahmen's contributions to both the theory of nonlinear approximation and to its application in a wide range of domains has been pervasive. Fortunately, the story is still going strong and I am happy to be going along for the ride.

References

1. I. Babuska and M. Vogelius, *Feedack and adaptive finite element solution of one dimensional boundary value problems*, Num. Math., **44**(1984), 75–102.
2. A. Barron, *Complexity regularization with application to artificial neural network*, in *Nonparametric functional estimation and related topics*, G. Roussas (ed.), 1990, 561–576, Kluwer Academic Publishers.
3. A. Barron, *Neural net approximation*, Proc 7th Yale Workshop on Adaptive and Learning Systems, K.S. Narendra Ed, New Haven, CT, 1992, pp. 69–72.
4. A. Barron, *Universal approximation bounds for superposition of n sigmoidal functions*, IEEE Trans. Inf. Theory, **39**(1993), 930–945.
5. A. Barron and G. Cheang *Penalized least squares, model selection, convex hull classes, and neural nets*, in Verleysen, M. (editor). Proceedings of the 9th ESANN, Brugge, Belgium, De-Facto press, 2001. pp. 371-376.
6. A. Barron, and G.H.L. Cheang *Risk bounds and greedy computations for penalized least squares, model selection, and neural networks* , Preprint, Department of Statistics, Yale University.
7. A. Barron, A. Cohen, W. Dahmen and R. DeVore, *Approximation and learning by greedy algorithms*, Annals of Statistics, **36**(2008), 64–94.
8. C. Bennett and R. Sharpley, *Interpolation of Operators*, in Pure and Applied Mathematics, 1988, Academic Press, N.Y.
9. P. Binev, A. Cohen, W. Dahmen, R. DeVore and V. Temlyakov, *Universal Algorithms for Learning Theory Part I: Piecewise Constant Functions*, J. Machine Learning, **6**(2005), 1297–1321.
10. P. Binev, W. Dahmen, and R. DeVore, *Adaptive Finite Element Methods with Convergence Rates*, Numerische Mathematik, **97**(2004), 219–268.
11. P. Binev, W. Dahmen, R. DeVore, and P. Petrushev, *Approximation Classes for Adaptive Methods*, Serdica Math. J., **28**(2002), 391–416.
12. P. Binev and R. DeVore, *Fast Computation in Adaptive Tree Approximation*, Num. Math., **97**(2004), 193–217.
13. M. Birman and M. Solomjak, *Piecewise polynomial approximations of functions of the classes* W_p^α, Mat. Sbornik, **73**(1967), 331–355.
14. E. Candès and T. Tao, *Decoding by linear programming*, IEEE Trans. Inf. Theory, **51**(2005), 4203–4215.
15. E. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure and Appl. Math., **59**(2006), 1207–1223.
16. A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods for elliptic operator equations: convergence rates*, Math. Comp., **70**(2000), 27–75.
17. A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods for operator equations: beyond the elliptic case*, J. FoCM, **2**(2002), 203–245.
18. A. Cohen, W. Dahmen and R. DeVore, *Compressed sensing and best k-term approximation*, J. Amer. Math. Soc., **22**(2009), 211–231.

19. A. Cohen, W. Dahmen and R. DeVore, *Instance Optimal Decoding by Thresholding in Compressed Sensing*, Contemporary Math., to appear.
20. A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, *Tree Approximation and Encoding*, ACHA, **11**(2001), 192–226.
21. A. Cohen, I. Daubechies, O. Guleryuz, and M. Orchard, *On the importance of combining wavelet based non-linear approximation in coding strategies*, IEEE Trans. Inf. Th., **48**(2002), 1895–1921.
22. G. Cormode and S. Muthukrishnan, *Towards an algorithmic theory of compressed sensing*, Technical Report 2005-25, DIMACS, 2005. Graham Cormode, S. Muthukrishnan: Combinatorial Algorithms for Compressed Sensing. SIROCCO 2006: 280–294.
23. S. Dahlke and R. DeVore, *Besov regularity for elliptic boundary value problems*, Communication in PDE's, **22**(1997), 1–16.
24. W. Dahmen and A. Kunoth, *Multilevel preconditioning*, Num. Math., **63**(1992), 315–344.
25. R. DeVore, *Nonlinear approximation*, Acta Numerica, **7**(1998), 51–150.
26. R. DeVore, B. Jawerth and B. Lucier, *Image compression through transform coding*, IEEE Proceedings on Information Theory, **38**(1992), 719–746.
27. R. DeVore, G. Kerkyacharian, D. Picard and V. Temlyakov, *On Mathematical Methods for Supervised Learning*, J. of FOCM, **6**(2006), 3–58.
28. R. DeVore, G. Petrova and P. Wojtaszczyk, *Instance-Optimality in Probability with an ℓ_1-minimization decoder*, ACHA, to appear.
29. R. DeVore and V. Temlyakov, *Some remarks on greedy algorithms*, Advances in Computational Mathematics, **5**(1996), 173–187.
30. D. Donoho, *Compressed Sensing*, IEEE Trans. Information Theory, **52**(2006), 1289–1306.
31. Donoho, D.L. and I.M. Johnstone, *Minimax Estimation via Wavelet shrinkage*, Annals of Statistics, **26**(1998), 879–921.
32. W. Dörfler, *A convergent adaptive scheme for Poisson's equation*, SIAM J. Num. Analysis, **33**(1996), 1106–1124.
33. B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, *Least angle regression*, Ann. Statist., **32**(2004), 407–499
34. A. Garnaev, E. Gluskin, *The widths of a Euclidean ball*, Doklady AN SSSR, **277**(1984), 1048–1052.
35. L. Györfy, M. Kohler, A. Krzyzak, and H. Walk, *A distribution-free theory of nonparametric regression*, 2002, Springer Verlag, Berlin.
36. T. Hastie, R. Tibshirani and J. Friedman *The Elements of Statistical Learning*, 2001, Springer.
37. H. Johnen and K. Scherer, *On the equivalence of the K-functional and moduli of continuity and some applications*, in *Constr. Theory of functions of several variables* , Proc. Conf. Oberwolfach 1976, Springer Lecture Notes 571), 119–140.
38. L. Jones, *A simple lemma on greedy approximation in Hilbert spaces and convergence rates for projection pursuit regression and neural network training*, Annals of Statistics, **20**(1992), 608–613.
39. B. Kashin, *The widths of certain finite dimensional sets and classes of smooth functions*, Izvestia, **41**(1977), 334–351.
40. S. Konyagin and V. Temlyakov, *Rate of convergence of Pure greedy Algorithm*, East J. Approx., **5**(1999), 493–499.
41. W. Lee, P. Bartlett and R. Williamson, *Efficient agnostic learning of neural networks with bounded fan-in*, IEEE Trans. Inf. Theory, **42**(1996), 2118–2132.
42. A. Litvak, A. Pajor, M. Rudelson, N. Tomczak-Jaegermann, *Smallest singular value of random matrices and geometry of random polytopes*, Advances in Math., **195**(2005), 491–523.
43. Livshitz, E.D. and V.N. Temlyakov, *Two lower estimates in greedy approximation*, Constr. Approx., **19**(2003), 509–524.
44. P. Morin, R. Nochetto, K. Siebert, *Data oscillation and convergence of adaptive FEM*, SIAM J. of Num. Anal., **38**(2000), 466–488.
45. D. Needell and J. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, ACHA, to appear.

46. D. Needell and R. Vershynin, *Uniform Uncertainty Principle and signal recovery via Regularized Orthogonal Matching Pursuit*, J. of FOCM, **9**(2009), 317–334.
47. D. Needell and R. Vershynin, *Signal Recovery from Inaccurate and Incomplete Measurements via Regularized Orthogonal Matching Pursuit*, preprint, 2007,
48. R. Stevenson, *Adaptive solution of operator equations using wavelet frames*, SIAM J. Num. Anal., **41**(2003), 1074–1100.
49. R. Stevenson, *An optimal adaptive finite element method*, SIAM J. Numer. Anal., **42**(2005), 2188–2217.
50. V. Temlyakov, *The best m-term approximation and greedy algorithms*, Adv. Comput. Math., **8**(1998), 249–265.
51. V. Temlyakov, *Nonlinear methods of approximation*, J. of FOCM, **3**(2003), 33–107.
52. R. Tibshirani, *Regression shrinkage and selection via the LASSO*, Journal of the Royal Statistical Society, Series B, **58**(1995), 267–288.
53. J. Tropp and A. Gilbert, *Signal recovery from random measurements via Orthogonal Matching Pursuit*, IEEE Trans. Info. Theory, **53**(2007), 4655–4666.
54. P. Wojtaszczyk, *Stability and instance optimality for Gaussian measurements in compressed sensing*, J. of FOCM, to appear

http://www.springer.com/978-3-642-03412-1

Multiscale, Nonlinear and Adaptive Approximation
Dedicated to Wolfgang Dahmen on the Occasion of his 60th Birthday
DeVore, R.; Kunoth, A. (Eds.)
2009, XXIV, 660 p., Hardcover
ISBN: 978-3-642-03412-1