From *SIAM News*, Volume 40, Number 8, October 2007

Approximation by Greedy Algorithms

By Albert Cohen

Approximation theory studies the process of approaching arbitrary functions by simple functions depending on N parameters, such as algebraic or trigonometric polynomials, finite elements, or wavelets. It plays a pivotal role in the analysis of numerical methods.

One usually makes the distinction between linear and nonlinear approximation. In the first case, the simple function is picked from a linear space (such as polynomials of degree N or piecewise-constant functions on some fixed partition of cardinality N) and is typically computed by projection of the arbitrary function onto this space. In the second case, the simple function is picked from a nonlinear space, yet is still characterizable by N parameters. Such a situation typically occurs with adaptive or data-driven approximations, which makes it relevant for applications as diverse as data compression, statistical estimation, or numerical schemes for partial differential or integral equations (see [4] for a general survey). The notion of projection is no longer applicable, however, and a critical question arises: *How can we compute the best possible approximation to a given function?* Let us translate this question into concrete terms for two specific examples:

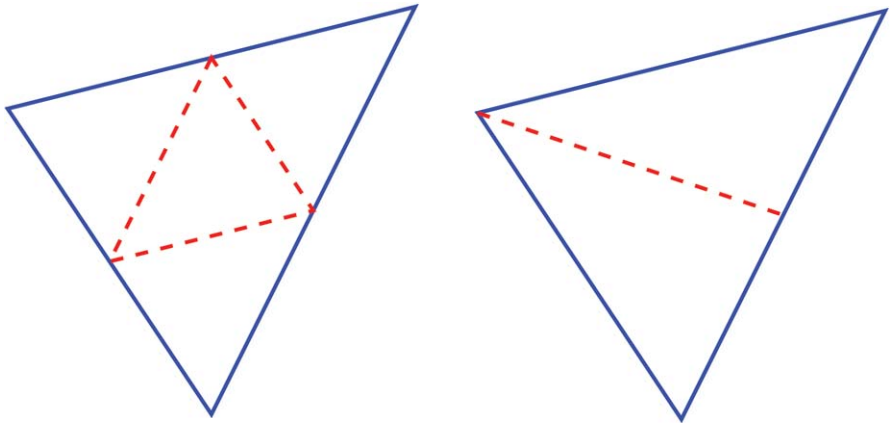

Figure 1. *Isotropic quad-split (left) and anisotropic bisection (right).*

■ **Adaptive triangulations.** Given a function f defined on a polygonal domain and given $N > 0$, find a partition of the domain into N triangles such that the L^2-error between f and its projection onto piecewise-polynomial functions of some fixed degree on this partition is minimized.

■ **Best N-term approximation.** Given a dictionary $\mathcal{D}$ of functions that is normalized and complete in some Hilbert space $\mathcal{H}$, and given $f \in \mathcal{H}$ and $N > 0$, find the combination $f_N = \sum_{k=1}^{N} c_k g_k$ that best approximates f, with $\{c_1, \cdots, c_N\}$ being real numbers and $\{g_1, \cdots, g_N\}$ picked from $\mathcal{D}$.

To make the first problem computationally tractable, we can assume that the vertices of each triangle are picked from a limited yet large number of locations M. For the second problem, our assumption is that the search is limited to a subset of $\mathcal{D}$ of cardinality M. The exhaustive search for the optimal solution has combinatorial complexity of order $\binom{M}{N}$, however, and neither problem is therefore generally solvable in polynomial time in N and M. A relevant goal then becomes to find suboptimal yet acceptable solutions that can be computed in reasonable time.

Greedy algorithms constitute a simple approach to this goal. They rely on stepwise local optimization procedures for picking the parameters in an inductive fashion, in the hope of approaching the globally optimal solution. They are particularly easy to im-plement, although the analysis of their performance gives rise to many open problems.

Triangulation problem. A simple greedy algorithm will typically start from a coarse triangulation and proceed from coarse to fine by splitting the triangle T on which the local projection error $\|f - P_T f\|_{L^2(T)}$ is maximal. The type of split can be fixed in advance, e.g., T can be decomposed from the mid-points into four subtriangles (Figure 1, left); alternatively, the split itself can be data driven, e.g., T can be bisected from one of its vertices, selected so as to minimize the local projection error for the new triangulation (Figure 1, right).

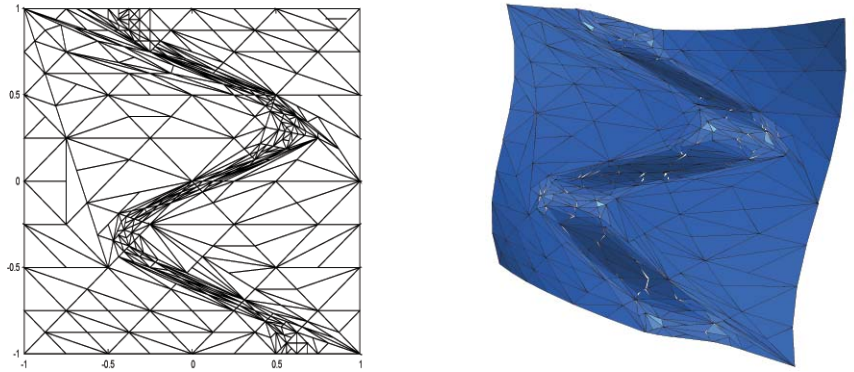

Figure 2. *Triangulation (left) and approximation (right).*

A split of either type restricts the accessible triangulations to a specific family. The second type has the advantage of allowing the development of anisotropic triangles, which are more efficient for approximating functions with curved singularities, such as edges in images or cliffs in terrain elevation data (see [3] for applications of this algorithm to image and surface compression). Figure 2 shows the result, after 512 steps, for the algorithm applied to the function $f(x) = y(x^2 + y^2) + \tanh(100(\sin(5y) - 2x))$, which has a sharp transition along the curve $\sin(5y) = 2x$.

Our algorithm behaves very well on this example, in the sense that it develops anisotrop-

ic triangles along the transition curve. But how close are we to an optimal triangulation? Because we have limited our choice to a restricted family, there is in general no hope that the greedy algorithm will produce precisely the optimal triangulation. In practice, we would be satisfied if we could show that the L^2 error between the function and its approximation decays with the number of triangles at a rate similar to that of the optimal one. Despite the good numerical behavior of the algorithm, however, no result of this type is known so far, and establishing the rate of convergence is a difficult task even for very specific functions.

N-term Approximation Problem. Greedy algorithms for solving problems of this type were introduced initially in the context of statistical data analysis. Their approximation properties were first explored in [1] and [6], in relation to neural network estimation, and in [5], for general dictionaries. A recent survey of such algorithms can be found in [7]. The most commonly used greedy algorithms are:

■ **Stepwise Projection (SP).** Having selected $\{g_1, \cdots, g_{k-1}\}$, we define f_{k-1} as the orthogonal projection onto $\mathrm{Span}\{g_1, \cdots, g_{k-1}\}$. The next g_k is selected so as to minimize the distance between f and $\mathrm{Span}\{g_1, \cdots, g_{k-1}, g\}$ among all choices of $g \in \mathcal{D}$.

■ **Orthonormal Matching Pursuit (OMP).** With the same projection for f_{k-1}, we select g_k so as to maximize the inner product $|\langle f - f_{k-1}, g\rangle|$ among all choices of $g \in \mathcal{D}$. Unlike the case with SP, we do not need to evaluate the anticipated projection error for all choices of $g \in \mathcal{D}$, which makes OMP more attractive from a computational viewpoint.

■ **Relaxed Greedy Algorithm (RGA).** Having constructed f_{k-1}, we define $f_k = \alpha_k f_{k-1} + \beta_k g_k$, where (α_k, β_k, g) are selected so as to minimize the distance between f and $\alpha f_{k-1} + \beta g$ among all choices of (α, β, g). It is often convenient to fix α_k in advance, which leads to the selection of g_k that maximizes $|\langle f - \alpha_k f_{k-1}, g\rangle|$ and $\beta_k = \langle f - \alpha_k f_{k-1}, g_k\rangle$. A typical choice is $\alpha_k = (1 - c/k)_+$ for some fixed $c > 1$. The intuitive role of the relaxation parameter α_k is to damp the memory of the algorithm, which might have been misled in its first steps. Because no orthogonal projection is involved, RGA is even cheaper than OMP.

In the special case in which the dictionary $\mathcal{D}$ is an orthonormal basis, SP and OMP are equivalent and provide the optimal solution to the best k-term approximation. We compute this solution simply by retaining the N largest coefficients $c_g = \langle f, g\rangle$ in the expansion of f, i.e., defining

$$f_N = \sum_{N \text{ largest}} |c_g| \, c_g g.$$

Intuitively, this approximation process is effective when the coefficient sequence $(c_g)_{g\in\mathcal{D}}$ is *concentrated* or *sparse*. One way to measure sparsity is to reorder the coefficients in decreasing order of magnitude and consider the smallest value of $0 < p \leq 2$ such that the resulting sequence $(c_n^*)_{n>0}$ decays like $n^{-1/p}$. We then say that the original coefficient sequence is weakly ℓ^p-summable (or belongs to $w\ell^p(\mathcal{D})$), with the extreme case $p = 0$ corresponding to a finitely supported sequence. We can easily check that this property is equivalent to the convergence rate

$$\|f - f_N\|_{\mathcal{H}} \leq CN^{-s}, \;\; s = 1/p - 1/2.$$

In summary, the convergence performance of the greedy algorithm is directly related to the level of sparsity of the coefficient sequence.

For a general dictionary $\mathcal{D}$, a natural question is whether a similar property holds: If f admits a sparse representation in $\mathcal{D}$, can we derive some corresponding rate of convergence for the greedy algorithm? A first answer to this question is given by the following result [5,6], which holds for SP, OMP, and RGA: If $f = \sum_{g\in\mathcal{D}} c_g g$ with $\|(c_g)\|_{\ell^1} \leq V$, then

$$\|f - f_N\|_{\mathcal{H}} \leq CV N^{-1/2}.$$

The case of a more general function $f \in \mathcal{H}$ that does not have a summable expansion can be treated by the following result [2], which again holds for SP, OMP, and RGA: If $f \in \mathcal{H}$ and $h = \sum_{g\in\mathcal{D}} d_g g$ with $\|(d_g)\|_{\ell^1} \leq W$, then

$$\|f - f_N\|_{\mathcal{H}} \leq \|f - h\|_{\mathcal{H}} + CW N^{-1/2}.$$

An immediate consequence is that the greedy algorithm converges for any $f \in \mathcal{H}$. This result means that the accuracy of the greedy approximant is stable under perturbation, although the component-selection process involved in the algorithm is unstable by nature.

Approaching the problem from the other end, one might ask how the algorithm behaves when f has a highly concentrated or finitely supported expansion, i.e., when $f = \sum_{g\in\mathcal{D}} c_g g$ with $\|(c_g)\|_{\ell^p} \leq V$ for some $p < 1$. For a general dictionary, it is known that SP, OMP, and RGA may fail to converge faster than $N^{-1/2}$. An area of active research is the study of those conditions on a dictionary $\mathcal{D}$ under which the convergence of greedy algorithms might fully benefit from such concentration properties, similar to the case of an orthonormal basis.

References

[1] A. Barron, *Universal approximation bounds for superposition of n sigmoidal functions*, IEEE Trans. Inf. Theory, 39 (1993), 930–945.

[2] A. Barron, A. Cohen, W. Dahmen, and R. DeVore, *Approximation and learning by greedy algorithms*, Ann. Stat., (2007), to appear.

[3] A. Cohen, N. Dyn, and F. Hecht, *A greedy triangulation algorithm for image and surface approximation and encoding*, preprint (2007).

[4] R. DeVore, *Nonlinear approximation*, Acta Num., 7 (1997), 51–150.

[5] R. DeVore and V. Temlyakov, *Some remarks on greedy algorithms*, Adv. Comput. Math., 5 (1996), 173–187.

[6] L.K. Jones, *A simple lemma on greedy approximation in Hilbert spaces and convergence rates for projection pursuit regression and neural network training*, Ann. Stat., 20 (1992), 608–613.

[7] V. Temlyakov, *Nonlinear methods of approximation*, J. FoCM, 3 (2003), 33–107.

Albert Cohen is a researcher in the Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, Paris.