INVITED PAPER

Computational Methods for Sparse Solution of Linear Inverse Problems

In many engineering areas, such as signal processing, practical results can be obtained by identifying approaches that yield the greatest quality improvement, or by selecting the most suitable computation methods.

By Joel A. Tropp, *Member IEEE*, and Stephen J. Wright

ABSTRACT | The goal of the sparse approximation problem is to approximate a target signal using a linear combination of a few elementary signals drawn from a fixed collection. This paper surveys the major practical algorithms for sparse approximation. Specific attention is paid to computational issues, to the circumstances in which individual methods tend to perform well, and to the theoretical guarantees available. Many fundamental questions in electrical engineering, statistics, and applied mathematics can be posed as sparse approximation problems, making these algorithms versatile and relevant to a plethora of applications.

KEYWORDS | Compressed sensing; convex optimization; matching pursuit; sparse approximation

I. INTRODUCTION

Linear inverse problems arise throughout engineering and the mathematical sciences. In most applications, these problems are ill-conditioned or underdetermined, so one must apply additional regularizing constraints in order to obtain interesting or useful solutions. Over the last two decades, sparsity constraints have emerged as a fundamental type of regularizer. This approach seeks an approximate solution to a linear system while requiring that the unknown has few nonzero entries relative to its dimension

$$\text{Find sparse } \boldsymbol{x} \quad \text{such that} \quad \boldsymbol{\Phi x} \approx \boldsymbol{u}$$

where $\boldsymbol{u}$ is a target signal and $\boldsymbol{\Phi}$ is a known matrix. Generically, this formulation is referred to as *sparse approximation* [1]. These problems arise in many areas, including statistics, signal processing, machine learning, coding theory, and approximation theory. *Compressive sampling* refers to a specific type of sparse approximation problem first studied in [2] and [3].

Tykhonov regularization, the classical device for solving linear inverse problems, controls the energy (i.e., the Euclidean norm) of the unknown vector. This approach leads to a linear least squares problem whose solution is generally nonsparse. To obtain sparse solutions, we must develop more sophisticated algorithms and often commit more computational resources. The effort pays off. Recent research has demonstrated that, in many cases of interest, there are algorithms that can find good solutions to large sparse approximation problems in reasonable time.

In this paper, we give an overview of algorithms for sparse approximation, describing their computational requirements and the relationships between them. We also discuss the types of problems for which each method is most effective in practice. Finally, we sketch the theoretical results that justify the application of these algorithms. Although low-rank regularization also falls within the sparse approximation framework, the algorithms we describe do not apply directly to this class of problems.

Section I-A describes "ideal" formulations of sparse approximation problems and some common features of

Manuscript received March 16, 2009; revised December 10, 2009; accepted February 11, 2010. Date of publication April 29, 2010; date of current version May 19, 2010. The work of J. A. Tropp was supported by the Office of Naval Research (ONR) under Grant N00014-08-1-2065. The work of S. J. Wright was supported by the National Science Foundation (NSF) under Grants CCF-0430504, DMS-0427689, CTS-0456694, CNS-0540147, and DMS-0914524.
J. A. Tropp is with the Applied and Computational Mathematics, Firestone Laboratories MC 217-50, California Institute of Technology, Pasadena, CA 91125-5000 USA (e-mail: jtropp@acm.caltech.edu).
S. J. Wright is with the Computer Sciences Department, University of Wisconsin, Madison, WI 53706 USA (e-mail: swright@cs.wisc.edu).
Digital Object Identifier: 10.1109/JPROC.2010.2044010

algorithms that attempt to solve these problems. Section II provides additional detail about greedy pursuit methods. Section III presents formulations based on convex programming and algorithms for solving these optimization problems.

A. Formulations

Suppose that $\boldsymbol{\Phi} \in \mathbb{R}^{m \times N}$ is a real matrix whose columns have unit Euclidean norm: $\|\varphi_j\|_2 = 1$ for $j = 1, 2, \ldots, N$. (The normalization does not compromise generality.) This matrix is often referred to as a *dictionary*. The columns of the matrix are "entries" in the dictionary, and a column submatrix is called a *subdictionary*.

The counting function $\|\cdot\|_0 : \mathbb{R}^N \to \mathbb{R}$ returns the number of nonzero components in its argument. We say that a vector $\boldsymbol{x}$ is *s-sparse* when $\|\boldsymbol{x}\|_0 \leq s$. When $\boldsymbol{u} = \boldsymbol{\Phi x}$, we refer to $\boldsymbol{x}$ as a *representation* of the signal $\boldsymbol{u}$ with respect to the dictionary.

In practice, signals tend to be *compressible*, rather than sparse. Mathematically, a compressible signal has a representation whose entries decay rapidly when sorted in order of decreasing magnitude. Compressible signals are well approximated by sparse signals, so the sparse approximation framework applies to this class. In practice, it is usually more challenging to identify approximate representations of compressible signals than of sparse signals.

The most basic problem we consider is to produce a maximally sparse representation of an observed signal $\boldsymbol{u}$

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{\Phi x} = \boldsymbol{u}. \tag{1}$$

One natural variation is to relax the equality constraint to allow some error tolerance $\varepsilon \geq 0$, in case the observed signal is contaminated with noise

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \|\boldsymbol{\Phi x} - \boldsymbol{u}\|_2 \leq \varepsilon. \tag{2}$$

It is most common to measure the prediction–observation discrepancy with the Euclidean norm, but other loss functions may also be appropriate.

The elements of (2) can be combined in several ways to obtain related problems. For example, we can seek the minimal error possible at a given level of sparsity $s \geq 1$

$$\min_{\boldsymbol{x}} \|\boldsymbol{\Phi x} - \boldsymbol{u}\|_2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_0 \leq s. \tag{3}$$

We can also use a parameter $\lambda > 0$ to balance the twin objectives of minimizing both error and sparsity

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{\Phi x} - \boldsymbol{u}\|_2^2 + \lambda \|\boldsymbol{x}\|_0. \tag{4}$$

If there are no restrictions on the dictionary $\boldsymbol{\Phi}$ and the signal $\boldsymbol{u}$, then sparse approximation is at least as hard as a general constraint satisfaction problem. Indeed, for fixed constants $C, K \geq 1$, it is NP-hard to produce a (Cs)-sparse approximation whose error lies within a factor K of the minimal s-term approximation error [4, Sec. 0.8.2].

Nevertheless, over the past decade, researchers have identified many interesting classes of sparse approximation problems that submit to computationally tractable algorithms. These striking results help to explain why sparse approximation has been such an important and popular topic of research in recent years.

In practice, sparse approximation algorithms tend to be slow unless the dictionary $\boldsymbol{\Phi}$ admits a fast matrix–vector multiply. Let us mention two classes of sparse approximation problems where this property holds. First, many naturally occurring signals are compressible with respect to dictionaries constructed using principles of harmonic analysis [5] (e.g., wavelet coefficients of natural images). This type of structured dictionary often comes with a fast transformation algorithm. Second, in compressive sampling, we typically view $\boldsymbol{\Phi}$ as the product of a random observation matrix and a fixed orthogonal matrix that determines a basis in which the signal is sparse. Again, fast multiplication is possible when both the observation matrix and sparsity basis are structured.

Recently, there have been substantial efforts to incorporate more sophisticated signal constraints into sparsity models. In particular, Baraniuk *et al.* have studied model-based compressive sampling algorithms, which use additional information such as the tree structure of wavelet coefficients to guide reconstruction of signals [6].

B. Major Algorithmic Approaches

There are at least five major classes of computational techniques for solving sparse approximation problems.

1) **Greedy pursuit**. Iteratively refine a sparse solution by successively identifying one or more components that yield the greatest improvement in quality [7].
2) **Convex relaxation**. Replace the combinatorial problem with a convex optimization problem. Solve the convex program with algorithms that exploit the problem structure [1].
3) **Bayesian framework**. Assume a prior distribution for the unknown coefficients that favors sparsity. Develop a maximum *a posteriori* estimator that incorporates the observation. Identify a region of significant posterior mass [8] or average over most-probable models [9].
4) **Nonconvex optimization**. Relax the ℓ_0 problem to a related nonconvex problem and attempt to identify a stationary point [10].
5) **Brute force**. Search through all possible support sets, possibly using cutting-plane methods to reduce the number of possibilities [11, Sec. 3.7–3.8].

This paper focuses on greedy pursuits and convex optimization. These two approaches are computationally practical and lead to provably correct solutions under well-defined conditions. Bayesian methods and nonconvex optimization are based on sound principles, but they do not currently offer theoretical guarantees. Brute force is, of course, algorithmically correct, but it remains plausible only for small-scale problems.

Recently, we have also seen interest in heuristic algorithms based on belief-propagation and message-passing techniques developed in the graphical models and coding theory communities [12], [13].

C. Verifying Correctness

Researchers have identified several tools that can be used to prove that sparse approximation algorithms produce optimal solutions to sparse approximation problems. These tools also provide insight into the *efficiency* of computational algorithms, so the theoretical background merits a summary.

The uniqueness of sparse representations is equivalent to an algebraic condition on submatrices of $\boldsymbol{\Phi}$. Suppose a signal $\boldsymbol{u}$ has two different s-sparse representations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Clearly

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}_1 = \boldsymbol{\Phi}\boldsymbol{x}_2 \quad \Longrightarrow \quad \boldsymbol{\Phi}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \boldsymbol{0}.$$

In other words, $\boldsymbol{\Phi}$ maps a nontrivial $(2s)$-sparse signal to zero. It follows that each s-sparse representation is unique if and only if each $(2s)$-column submatrix of $\boldsymbol{\Phi}$ is injective.

To ensure that sparse approximation is computationally tractable, we need stronger assumptions on $\boldsymbol{\Phi}$. Not only should sparse signals be *uniquely* determined, but they should be *stably* determined. Consider a signal perturbation $\Delta\boldsymbol{u}$ and an s-sparse coefficient perturbation $\Delta\boldsymbol{x}$, related by $\Delta\boldsymbol{u} = \boldsymbol{\Phi}(\Delta\boldsymbol{x})$. Stability requires that $\|\Delta\boldsymbol{x}\|_2$ and $\|\Delta\boldsymbol{u}\|_2$ are comparable.

This property is commonly imposed by fiat. We say that the matrix $\boldsymbol{\Phi}$ satisfies the restricted isometry property (RIP) of order K with constant $\delta = \delta_K < 1$ if

$$\|\boldsymbol{x}\|_0 \leq K \Rightarrow (1-\delta)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2. \quad (5)$$

For sparse approximation, we hope (5) holds for large K. This concept was introduced in the important paper [14]; some refinements appear in [15].

The RIP can be verified using the *coherence statistic* of the matrix $\boldsymbol{\Phi}$, which is defined as

$$\mu = \max_{j \neq k} \left| \langle \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k \rangle \right|.$$

An elementary argument [16] via Gershgorin's circle theorem establishes that the RIP constant $\delta_K \leq \mu(K-1)$. In signal processing applications, it is common that $\mu \approx m^{-1/2}$, so we have nontrivial RIP bounds for $K \approx \sqrt{m}$. Unfortunately, no known deterministic matrix yields a substantially better RIP. Early references for coherence include [7] and [17].

Certain random matrices, however, satisfy much stronger RIP bounds with high probability. For Gaussian and Bernoulli matrices, RIP holds when $K \approx m/\log(N/m)$. For more structured matrices, such as a random section of a discrete Fourier transform, RIP often holds when $K \approx m/\log^p(N)$ for a small integer p. This fact explains the benefit of randomness in compressive sampling. Establishing the RIP for a random matrix requires techniques more sophisticated than the simple coherence arguments; see [14] for discussion.

Recently, researchers have observed that sparse matrices may satisfy a related property, called RIP-1, even when they do not satisfy (5). RIP-1 can also be used to analyze sparse approximation algorithms. Details are given in [18].

D. Cross-Cutting Issues

Structural properties of the matrix $\boldsymbol{\Phi}$ have a substantial impact on the implementation of sparse approximation algorithms. In most applications of interest, the large size or lack of sparseness in $\boldsymbol{\Phi}$ makes it impossible to store this matrix (or any substantial submatrix) explicitly in computer memory. Often, however, matrix–vector products involving $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ can be performed efficiently. For example, the cost of these products is $O(N \log N)$ when $\boldsymbol{\Phi}$ is constructed from Fourier or wavelet bases. For algorithms that solve least squares problems, a fast multiply is particularly important because it allows us to use iterative methods such as LSQR or conjugate gradient (CG). In fact, all the algorithms discussed below can be implemented in a way that requires access to $\boldsymbol{\Phi}$ only through matrix–vector products.

Spectral properties of subdictionaries, such as those encapsulated in (5), have additional implications for the computational cost of sparse approximation algorithms. Some methods exhibit fast linear asymptotic convergence because the RIP ensures that the subdictionaries encountered during execution have superb conditioning. Other approaches (for example, interior-point methods) are less sensitive to spectral properties, so they become more competitive when the RIP is less pronounced or the target signal is not particularly sparse.

It is worth mentioning here that most algorithmic papers in sparse reconstruction present computational results only on synthetic test problems. Test problem collections representative of sparse approximation problems encountered in practice are crucial to guiding further development of algorithms. A significant effort in this direction is Sparco [19], a Matlab environment for interfacing algorithms and constructing test problems that also includes a variety of problems gathered from the literature.

II. PURSUIT METHODS

A *pursuit method* for sparse approximation is a greedy approach that iteratively refines the current estimate for the coefficient vector $\boldsymbol{x}$ by modifying one or several coefficients chosen to yield a substantial improvement in approximating the signal. We begin by describing the simplest effective greedy algorithm, orthogonal matching pursuit (OMP), and summarizing its theoretical guarantees. Afterward, we outline a more sophisticated class of modern pursuit techniques that has shown promise for compressive sampling problems. We briefly discuss iterative thresholding methods, and conclude with some general comments about the role of greedy algorithms in sparse approximation.

A. Orthogonal Matching Pursuit

OMP is one of the earliest methods for sparse approximation. Basic references for this method in the signal processing literature are [20] and [21], but the idea can be traced to 1950s work on variable selection in regression [11].

Fig. 1 contains a mathematical description of OMP. The symbol $\boldsymbol{\Phi}_\Omega$ denotes the subdictionary indexed by a subset Ω of $\{1, 2, \ldots, N\}$.

In a typical implementation of OMP, the identification step is the most expensive part of the computation. The most direct approach computes the maximum inner product via the matrix–vector multiplication $\boldsymbol{\Phi}^* \boldsymbol{r}_{k-1}$, which costs $\mathrm{O}(mN)$ for an unstructured dense matrix. Some authors have proposed using nearest neighbor data structures to perform the identification query more efficiently [22]. In certain applications, such as projection pursuit regression, the "columns" of $\boldsymbol{\Phi}$ are indexed by a continuous parameter, and identification can be posed as a low-dimensional optimization problem [23].

The estimation step requires the solution of a least squares problem. The most common technique is to maintain a QR factorization of $\boldsymbol{\Phi}_{\Omega_k}$, which has a marginal cost of $\mathrm{O}(mk)$ in the kth iteration. The new residual $\boldsymbol{r}_k$ is a by-product of the least squares problem, so it requires no extra computation.

There are several natural stopping criteria.

- Halt after a fixed number of iterations: $k = s$.
- Halt when the residual has small magnitude: $\|\boldsymbol{r}_k\|_2 \leq \varepsilon$.
- Halt when no column explains a significant amount of energy in the residual: $\|\boldsymbol{\Phi}^* \boldsymbol{r}_{k-1}\|_\infty \leq \varepsilon$.

These criteria can all be implemented at minimal cost.

Many related greedy pursuit algorithms have been proposed in the literature; we cannot do them all justice here. Some particularly noteworthy variants include matching pursuit [7], the relaxed greedy algorithm [24], and the ℓ_1-penalized greedy algorithm [25].

B. Guarantees for Simple Pursuits

OMP produces the residual $\boldsymbol{r}_m = \boldsymbol{0}$ after m steps (provided that the dictionary can represent the signal $\boldsymbol{u}$ exactly), but this representation hardly qualifies as sparse. Classical analyses of greedy pursuit focus instead on the rate of convergence.

Greedy pursuits often converge linearly with a rate that depends on how well the dictionary covers the sphere [7]. For example, OMP offers the estimate

$$\|\boldsymbol{r}_k\|_2 \leq (1 - \varrho^2)^{k/2} \|\boldsymbol{u}\|_2$$

where

$$\varrho = \inf_{\|\boldsymbol{v}\|_2 = 1} \sup_n |\langle \boldsymbol{v}, \boldsymbol{\varphi}_n \rangle|.$$

(See [21, Sec. 3] for details.) Unfortunately, the covering parameter ϱ is typically $\mathrm{O}(m^{-1/2})$ unless the number N of atoms is huge, so this estimate has limited interest.

A second type of result demonstrates that the rate of convergence depends on how well the dictionary expresses the signal of interest [24, eq. (1.9)]. For example, OMP offers the estimate

$$\|\boldsymbol{r}_k\|_2 \leq k^{-1/2} \|\boldsymbol{u}\|_{\boldsymbol{\Phi}}$$

where

$$\|\boldsymbol{u}\|_{\boldsymbol{\Phi}} = \inf\{\|\boldsymbol{x}\|_1 : \boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}\}.$$

- **Input.** A signal $\boldsymbol{u} \in \mathbb{R}^m$, a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times N}$
- **Output.** A sparse coefficient vector $\boldsymbol{x} \in \mathbb{R}^N$

1) **Initialize.** Set the index set $\Omega_0 = \emptyset$, the residual $\boldsymbol{r}_0 = \boldsymbol{u}$, and put the counter $k = 1$.
2) **Identify.** Find a column n_k of $\boldsymbol{\Phi}$ that is most strongly correlated with the residual:
$$n_k \in \arg\max_n |\langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_n \rangle| \quad \text{and} \quad \Omega_k = \Omega_{k-1} \cup \{n_k\}.$$
3) **Estimate.** Find the best coefficients for approximating the signal with the columns chosen so far.
$$\boldsymbol{x}_k = \arg\min_{\boldsymbol{y}} \|\boldsymbol{u} - \boldsymbol{\Phi}_{\Omega_k} \boldsymbol{y}\|_2.$$
4) **Iterate.** Update the residual:
$$\boldsymbol{r}_k = \boldsymbol{u} - \boldsymbol{\Phi}_{\Omega_k} \boldsymbol{x}_k.$$
Increment k. Repeat (2)–(4) until stopping criterion holds.
5) **Output.** Return the vector $\boldsymbol{x}$ with components $x(n) = x_k(n)$ for $n \in \Omega_k$ and $x(n) = 0$ otherwise.

Fig. 1. *Orthogonal matching pursuit.*

The dictionary norm $\|\cdot\|_{\mathbf{\Phi}}$ is typically small when its argument has a good sparse approximation. For further improvements on this estimate, see [26]. This bound is usually superior to the exponential rate estimate above, but it can be disappointing for signals with excellent sparse approximations.

Subsequent work established that greedy pursuit produces near-optimal sparse approximations with respect to *incoherent* dictionaries [22], [27]. For example, if $3\mu k \leq 1$, then

$$\|\boldsymbol{r}_k\|_2 \leq \sqrt{1+6k}\,\|\boldsymbol{u}-\boldsymbol{a}_k^\star\|_2$$

where $\boldsymbol{a}_k^\star$ denotes the best ℓ_2 approximation of $\boldsymbol{u}$ as a linear combination of k columns from $\mathbf{\Phi}$. See [28]–[30] for refinements.

Finally, when $\mathbf{\Phi}$ is sufficiently random, OMP provably recovers s-sparse signals when $s \leq m/(2\log N)$ and the parameters are sufficiently large [31], [32].

C. Contemporary Pursuit Methods

For many applications, OMP does not offer adequate performance, so researchers have developed more sophisticated pursuit methods that work better in practice and yield essentially optimal theoretical guarantees. These techniques depend on several enhancements to the basic greedy framework:

1) selecting multiple columns per iteration;
2) pruning the set of active columns at each step;
3) solving the least squares problems iteratively;
4) theoretical analysis using the RIP bound (5).

Although modern pursuit methods were developed specifically for compressive sampling problems, they also offer attractive guarantees for sparse approximation.

There are many early algorithms that incorporate some of these features. For example, stagewise orthogonal matching pursuit (StOMP) [33] selects multiple columns at each step. The regularized orthogonal matching pursuit algorithm [34], [35] was the first greedy technique whose analysis was supported by a RIP bound (5). For historical details, we refer the reader to the discussion in [36, Sec. 7].

Compressive sampling matching pursuit (CoSaMP) [36] was the first algorithm to assemble these ideas to obtain essentially optimal performance guarantees. Dai and Milenkovic describe a similar algorithm, called subspace pursuit, with equivalent guarantees [37]. Other natural variants are described in [38, App. A.2]. Because of space constraints, we focus on the CoSaMP approach.

Fig. 2 describes the basic CoSaMP procedure. The notation $[\boldsymbol{x}]_r$ denotes the restriction of a vector $\boldsymbol{x}$ to the r components largest in magnitude (ties broken lexicographically), while $\operatorname{supp}(\boldsymbol{x})$ denotes the support of the vector $\boldsymbol{x}$, i.e., the set of nonzero components. The natural value for the tuning parameter is $\alpha = 1$, but empirical refinement may be valuable in applications [39].

- **Input.** A signal $\boldsymbol{u} \in \mathbb{R}^m$, a matrix $\mathbf{\Phi} \in \mathbb{R}^{m\times N}$, target sparsity s, tuning parameter α.
- **Output.** An s-sparse coefficient vector $\boldsymbol{x} \in \mathbb{R}^N$

1) **Initialize.** Set the initial coefficient vector $\boldsymbol{x}_0 = \mathbf{0}$ and the residual $\boldsymbol{r}_0 = \boldsymbol{u}$. Let $k = 1$.
2) **Identify.** Find αs columns of $\mathbf{\Phi}$ that are most strongly correlated with the residual:
$$\Omega \in \arg\min_{|T|\leq \alpha s} \sum\nolimits_{n\in T} |\langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_n\rangle|.$$
3) **Merge.** Put the old and new columns into one set:
$$T = \operatorname{supp}(\boldsymbol{x}_{k-1}) \cup \Omega$$
4) **Estimate.** Find the best coefficients for approximating the residual with these columns:
$$\boldsymbol{y}_k = \arg\min_{\boldsymbol{y}} \|\boldsymbol{r}_{k-1} - \mathbf{\Phi}_T \boldsymbol{y}\|_2$$
5) **Prune.** Retain the s largest coefficients:
$$\boldsymbol{x}_k = [\boldsymbol{y}]_s.$$
6) **Iterate.** Update the residual:
$$\boldsymbol{r}_k = \boldsymbol{u} - \mathbf{\Phi}\boldsymbol{x}_k.$$
Repeat (2)–(5) until stopping criterion holds.
7) **Output.** Return $\boldsymbol{x} = \boldsymbol{x}_k$.

Fig. 2. ***Compressive sampling matching pursuit.***

Both the practical performance and theoretical analysis of CoSaMP require the dictionary $\mathbf{\Phi}$ to satisfy the RIP (5) of order $2s$ with constant $\delta_{2s} \ll 1$. Of course, these methods can be applied without the RIP, but the behavior is unpredictable. A heuristic for identifying the maximum sparsity level s is to require that $s \leq m/(2\log(1+N/s))$.

Under the RIP hypothesis, each iteration of CoSaMP reduces the approximation error by a constant factor until it approaches its minimal value. To be specific, suppose that the signal $\boldsymbol{u}$ satisfies

$$\boldsymbol{u} = \mathbf{\Phi}\boldsymbol{x}^\star + \boldsymbol{e} \tag{6}$$

for unknown coefficient vector $\boldsymbol{x}^\star$ and noise term $\boldsymbol{e}$. If we run the algorithm for a sufficient number of iterations, the output $\boldsymbol{x}$ satisfies

$$\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2 \leq Cs^{-1/2}\left\|\boldsymbol{x}^\star - [\boldsymbol{x}^\star]_{s/2}\right\|_1 + C\|\boldsymbol{e}\|_2 \tag{7}$$

where C is a constant. The form of this error bound is optimal [40].

Stopping criteria are tailored to the signals of interest. For example, when the coefficient vector $\boldsymbol{x}^\star$ is compressible, the algorithm requires only $\mathrm{O}(\log N)$ iterations. Under the RIP hypothesis, each iteration requires a constant number of multiplications with $\mathbf{\Phi}$ and $\mathbf{\Phi}^*$ to solve the least squares

problem. Thus, the total running time is $O(N \log^2 N)$ for a structured dictionary and a compressible signal.

In practice, CoSaMP is faster and more effective than OMP for compressive sampling problems, except perhaps in the ultrasparse regime where the number of nonzeros in the representation is very small. CoSaMP is faster but usually less effective than algorithms based on convex programming.

D. Iterative Thresholding

Modern pursuit methods are closely related to iterative thresholding algorithms, which have been studied extensively over the last decade. (See [39] for a current bibliography.) Section III-D describes additional connections with optimization-based approaches.

Among thresholding approaches, iterative hard thresholding (IHT) is the simplest. It seeks an s-sparse representation $\boldsymbol{x}_\star$ of a signal $\boldsymbol{u}$ via the iteration

$$\begin{cases} \boldsymbol{x}_0 = \boldsymbol{0} \\ \boldsymbol{r}_k = \boldsymbol{u} - \boldsymbol{\Phi}\boldsymbol{x}_k \\ \boldsymbol{x}_{k+1} = [\boldsymbol{x}_k + \boldsymbol{\Phi}^* \boldsymbol{r}_k]_s, \quad k \geq 0. \end{cases}$$

Blumensath and Davies [41] have established that IHT admits an error guarantee of the form (7) under a RIP hypothesis of the form $\delta_{2s} \ll 1$. For related results on IHT, see [42]. Garg and Khandekar [43] describe a similar method, gradient descent with sparsification, and present an elegant analysis, which is further simplified in [44].

There is empirical evidence that thresholding is reasonably effective for solving sparse approximation problems in practice; see, e.g., [45]. On the other hand, some simulations indicate that simple thresholding techniques behave poorly in the presence of noise [41, Sec. 8].

Very recently, Donoho and Maliki have proposed a more elaborate method, called two-stage thresholding (TST) [39]. They describe this approach as a hybrid of CoSaMP and thresholding, modified with extra tuning parameters. Their work includes extensive simulations meant to identify optimal parameter settings for TST. By construction, these optimally tuned algorithms dominate related approaches with fewer parameters. The discussion in [39] focuses on perfectly sparse, random signals, so the applicability of the approach to signals that are compressible, noisy, or deterministic is unclear.

E. Commentary

Greedy pursuit methods have often been considered naive, in part because there are contrived examples where the approach fails spectacularly; see [1, Sec. 2.3.2]. However, recent research has clarified that greedy pursuits succeed empirically and theoretically in many situations where convex relaxation works. In fact, the boundary between greedy methods and convex relaxation methods is somewhat blurry. The greedy selection technique is closely related to dual coordinate-ascent algorithms, while certain methods for convex relaxation, such as least-angle regression [46] and homotopy [47], use a type of greedy selection at each iteration. We can make certain general observations, however. Greedy pursuits, thresholding, and related methods (such as homotopy) can be quite fast, especially in the ultrasparse regime. Convex relaxation algorithms are more effective at solving sparse approximation problems in a wider variety of settings, such as those in which the signal is not very sparse and heavy observational noise is present.

Greedy techniques have several additional advantages that are important to recognize. First, when the dictionary contains a continuum of elements (as in projection pursuit regression), convex relaxation may lead to an infinite-dimensional primal problem, while the greedy approach reduces sparse approximation to a sequence of simple 1-D optimization problems. Second, greedy techniques can incorporate constraints that do not fit naturally into convex programming formulations. For example, the data stream community has proposed efficient greedy algorithms for computing near-optimal histograms and wavelet-packet approximations from compressive samples [4]. More recently, it has been shown that CoSaMP can be modified to enforce tree-like constraints on wavelet coefficients. Extensions to simultaneous sparse approximation problems have also been developed [6]. This is an exciting and important line of work.

At this point, it is not fully clear what role greedy pursuit algorithms will ultimately play in practice. Nevertheless, this strand of research has led to new tools and insights for analyzing other types of algorithms for sparse approximation, including the iterative thresholding and model-based approaches above.

III. OPTIMIZATION

Another fundamental approach to sparse approximation replaces the combinatorial ℓ_0 function in the mathematical programs from Section I-A with the ℓ_1-norm, yielding *convex* optimization problems that admit tractable algorithms. In a concrete sense [48], the ℓ_1-norm is the closest convex function to the ℓ_0 function, so this "relaxation" is quite natural.

The convex form of the equality-constrained problem (1) is

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{u} \tag{8}$$

while the mixed formulation (4) becomes

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{u}\|_2^2 + \tau\|\boldsymbol{x}\|_1. \tag{9}$$

Here, $\tau \geq 0$ is a *regularization parameter* whose value governs the sparsity of the solution: large values typically produce sparser results. It may be difficult to select an appropriate value for τ in advance, since it controls the sparsity indirectly. As a consequence, we often need to solve (9) repeatedly for different choices of this parameter, or to trace systematically the path of solutions as τ decreases toward zero. When $\tau \geq \|\Phi^* \boldsymbol{u}\|_\infty$, the solution of (9) is $\boldsymbol{x} = \boldsymbol{0}$.

Another variant is the least absolute shrinkage and selection operator formulation [49], which first arose in the context of variable selection

$$\min_{\boldsymbol{x}} \|\Phi \boldsymbol{x} - \boldsymbol{u}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_1 \leq \beta. \tag{10}$$

The LASSO is equivalent to (9) in the sense that the path of solutions to (10) parameterized by positive β matches the solution path for (9) as τ varies. Finally, we note another common formulation

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|\Phi \boldsymbol{x} - \boldsymbol{u}\|_2 \leq \varepsilon \tag{11}$$

that explicitly parameterizes the error norm.

A. Guarantees

It has been demonstrated that convex relaxation methods produce optimal or near-optimal solutions to sparse approximation problems in a variety of settings.

The earliest results [16], [17], [27] establish that the equality-constrained problem (8) correctly recovers all s-sparse signals from an incoherent dictionary provided that $2\mu s \leq 1$. In the best case, this bound applies at the sparsity level $s \approx \sqrt{m}$. Subsequent work [29], [50], [51] showed that the convex programs (9) and (11) can identify noisy sparse signals in a similar parameter regime.

The results described above are sharp for deterministic signals, but they can be extended significantly for *random* signals that are sparse with respect to an incoherent dictionary. The paper [52] proves that the equality-constrained problem (8) can identify random signals, even when the sparsity level s is approximately $m/\log m$. Most recently, the paper [53] observed that ideas from [51] and [54] imply that the convex relaxation (9) can identify noisy, random sparse signals in a similar parameter regime.

Results from [14] and [55] demonstrate that convex relaxation succeeds well in the presence of the RIP. Suppose that signal $\boldsymbol{u}$ and unknown coefficient vector $\boldsymbol{x}^\star$ are related as in (6) and that the dictionary Φ has RIP constant $\delta_{2s} \ll 1$. Then, the solution $\boldsymbol{x}$ to (11) verifies

$$\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2 \leq Cs^{-1/2}\|\boldsymbol{x}^\star - [\boldsymbol{x}^\star]_s\|_1 + C\varepsilon$$

for some constant C, provided that $\varepsilon \geq \|\boldsymbol{e}\|_2$. Compare this bound with the error estimate (7) for CoSaMP and IHT.

An alternative approach for analyzing convex relaxation algorithms relies on geometric properties of the kernel of the dictionary [40], [56]–[58]. Another geometric method, based on random projections of standard polytopes, is studied in [59] and [60].

B. Active Set/Pivoting

Pivoting algorithms explicitly trace the path of solutions as the scalar parameter in (10) ranges across an interval. These methods exploit the piecewise linearity of the solution as a function of β, a consequence of the fact that the optimality Karush–Kuhn–Tucker (KKT) conditions can be stated as a linear complementarity problem. By referring to the KKT system, we can quickly identify the next "breakpoint" on the solution path—the nearest value of β at which the derivative of the piecewise-linear function changes.

The homotopy method of [47] follows this approach. It starts with $\beta = 0$, where the solution of (10) is $\boldsymbol{x} = \boldsymbol{0}$, and it progressively locates the next largest value of β where a component of $\boldsymbol{x}$ switches from a zero to a nonzero, or *vice versa*. At each step, the method updates or downdates a QR factorization of the submatrix of Φ that corresponds to the nonzero components of $\boldsymbol{x}$. A similar method [46] is implemented as `SolveLasso` in the SparseLab toolbox.[1] Related approaches can be developed for the formulation (9).

If we limit our attention to values of β for which $\boldsymbol{x}$ has few nonzeros, the active-set/pivoting approach is efficient. The homotopy method requires about $2s$ matrix–vector multiplications by Φ or Φ^*, to identify s nonzeros in $\boldsymbol{x}$, together with $O(ms^2)$ operations for updating the factorization and performing other linear algebra operations. This cost is comparable with OMP.

OMP and homotopy are quite similar in that the solution is altered by systematically adding nonzero components to $\boldsymbol{x}$ and updating the solution of a reduced linear least squares problem. In each case, the criterion for selecting components involves the inner products between inactive columns of Φ and the residual $\boldsymbol{u} - \Phi \boldsymbol{x}$. One notable difference is that homotopy occasionally allows for nonzero components of $\boldsymbol{x}$ to return to zero status. See [46] and [61] for other comparisons.

C. Interior-Point Methods

Interior-point methods were among the first approaches developed for solving sparse approximation problems by convex optimization. The early algorithms [1], [62] apply a primal–dual interior-point framework where the innermost subproblems are formulated as linear least squares problems that can be solved with iterative methods, thus allowing these methods to take advantage

[1] http://sparselab.stanford.edu.

of fast matrix–vector multiplications involving Φ and $\Phi^\star$. An implementation is available as `pdco` and `SolveBP` in the SparseLab toolbox.

Other interior-point methods have been proposed expressly for compressive sampling problems. The paper [63] describes a primal log-barrier approach for a quadratic programming reformulation of (9):

$$\min_{x,z} \frac{1}{2}\|\Phi x - u\|_2^2 + \tau \mathbf{1}^T z \quad \text{subject to} \quad -z \le x \le z.$$

The technique relies on a specialized preconditioner that allows the internal Newton iterations to be completed efficiently with CG. The method[2] is implemented as the code `l1_ls`. The ℓ_1-magic package[3] [64] contains a primal log-barrier code for the second-order cone formulation (11), which includes the option of solving the innermost linear system with CG.

In general, interior-point methods are not competitive with the gradient methods of Section III-D on problems with very sparse solutions. On the other hand, their performance is insensitive to the sparsity of the solution or the value of the regularization parameter. Interior-point methods can be robust in the sense that there are not many cases of very slow performance or outright failure, which sometimes occurs with other approaches.

D. Gradient Methods

Gradient-descent methods, also known as *first-order* methods, are iterative algorithms for solving (9) in which the major operation at each iteration is to form the gradient of the least squares term at the current iterate, viz., $\Phi^*(\Phi x_k - u)$. Many of these methods compute the next iterate x_{k+1} using the rules

$$x_k^+ := \arg\min_z (z - x_k)^* \Phi^*(\Phi x_k - u) + \frac{1}{2}\alpha_k \|z - x_k\|_2^2 + \tau \|z\|_1 \quad (12a)$$

$$x_{k+1} := x_k + \gamma_k (x_k^+ - x_k) \quad (12b)$$

for some choice of scalar parameters α_k and γ_k. Alternatively, we can write subproblem (12a) as

$$x_k^+ := \arg\min_z \frac{1}{2}\left\| z - \left(x_k - \frac{1}{\alpha_k}\Phi^*(\Phi x_k - u)\right)\right\|_2^2 + \frac{\tau}{\alpha_k}\|z\|_1. \quad (13)$$

[2]www.stanford.edu/~boyd/l1_ls/.
[3]www.l1-magic.org.

- **Input.** A signal $u \in \mathbb{R}^m$, a matrix $\Phi \in \mathbb{R}^{m\times N}$, regularization parameter $\tau > 0$, initial estimate x_0 of the representation vector.
- **Output.** Coefficient vector $x \in \mathbb{R}^N$

1) **Initialize.** Set $k = 1$.
2) **Iterate.** Choose $\alpha_k \ge 0$ and obtain x_k^+ from (12a). If an acceptance test on x_k^+ is not passed, increase α_k by some factor and repeat.
3) **Line Search.** Choose $\gamma_k \in (0, 1]$ and obtain x_{k+1} from (12b).
4) **Test.** If stopping criterion holds, terminate with $x = x_{k+1}$. Otherwise, set $k \leftarrow k + 1$ and go to (2).

Fig. 3. *Gradient-descent framework.*

Algorithms that compute steps of this type are known by such labels as operator splitting [65], iterative splitting and thresholding (IST) [66], fixed-point iteration [67], and sparse reconstruction via separable approximation (SpaRSA) [68]. Fig. 3 shows the framework for this class of methods.

Standard convergence results for these methods, e.g., [65, Th. 3.4], require that $\inf_k \alpha_k > \|\Phi^*\Phi\|_2/2$, a tight restriction that leads to slow convergence in practice. The more practical variants described in [68] admit smaller values of α_k, provided that a sufficient decrease in the objective in (9) occurs over a span of successive iterations. Some variants use Barzilai–Borwein formulas that select values of α_k lying in the spectrum of $\Phi^*\Phi$. When x_k^+ fails the acceptance test in Step 2, the parameter α_k is increased (repeatedly, as necessary) by a constant factor. Step lengths $\gamma_k \equiv 1$ are used in [67] and [68]. The iterated hard shrinkage method of [69] sets $\alpha_k \equiv 0$ in (12) and chooses γ_k to do a conditional minimization along the search direction.

Related approaches include TwIST [70], a variant of IST that is significantly faster in practice, and which deviates from the framework of Fig. 3 in that the previous iterate x_{k-1} also enters into the step calculation (in the manner of successive over-relaxation approaches for linear equations). GPSR [71] is simply a gradient-projection algorithm for the convex quadratic program obtained by splitting x into positive and negative parts.

The approaches above tend to work well on sparse signals when the dictionary Φ satisfies the RIP. Often, the nonzero components of x are identified quickly, after which the method reduces essentially to an iterative method for the reduced linear least squares problem in these components. Because of the RIP, the active submatrix is well conditioned, so these final iterates converge quickly. In fact, these steps are quite similar to the estimation step of CoSaMP.

These methods benefit from warm starting, that is, the work required to identify a solution can be reduced dramatically when the initial estimate x_0 in Step 1 is close to the solution. This property can be used to ameliorate the

often poor performance of these methods on problems for which (9) is not particularly sparse or the regularization parameter τ is small. *Continuation* strategies have been proposed for such cases, in which we solve (9) for a decreasing sequence of values of τ, using the approximate solution for each value as the starting point for the next subproblem. Continuation can be viewed as a coarse-grained, approximate variant of the pivoting strategies of Section III-B, which track individual changes in the active components of $\boldsymbol{x}$ explicitly. Some continuation methods are described in [67] and [68]. Though adaptive strategies for choosing the decreasing sequence of τ values have been proposed, the design of a robust, practical, and theoretically effective continuation algorithm remains an interesting open question.

E. Extensions of Gradient Methods

Second-order information can be used to enhance gradient projection approaches by taking approximate reduced Newton steps in the subset of components of $\boldsymbol{x}$ that appears to be nonzero. In some approaches [68], [71], this enhancement is made only after the first-order algorithm is terminated as a means of removing the bias in the formulation (9) introduced by the regularization term. Other methods [72] apply this technique at intermediate steps of the algorithm. (A similar approach was proposed for the related problem of ℓ_1-regularized logistic regression in [73].) Iterative methods such as conjugate gradient can be used to find approximate solutions to the reduced linear least squares problems. These subproblems are, of course, closely related to the ones that arise in the greedy pursuit algorithms of Section II.

The SPG method of [74, Sec. 4] applies a different type of gradient projection to the formulation (10). This approach takes steps along the negative gradient of the least squares objective in (10), with steplength chosen by a Barzilai–Borwein formula (with backtracking to enforce sufficient decrease over a reference function value), and projects the resulting vector onto the constraint set $\|\boldsymbol{x}\|_1 \leq \beta$. Since the ultimate goal in [74] is to solve (11) for a given value of ε, the approach above is embedded into a scalar equation solver that identifies the value of β for which the solution of (10) coincides with the solution of (11).

An important recent line of work has involved applying optimal gradient methods for convex minimization [75]–[77] to the formulations (9) and (11). These methods have many variants, but they share the goal of finding an approximate solution that is as close as possible to the optimal set (as measured by norm-distance or by objective value) in a given budget of iterations. (By contrast, most iterative methods for optimization aim to make significant progress during each individual iteration.) Optimal gradient methods typically generate several concurrent sequences of iterates, and they have complex steplength rules that depend on some prior knowledge, such as the Lipschitz constant of the gradient. Specific works that apply optimal gradient methods to sparse approximation include [78]–[80]. These methods may perform better than simple gradient methods when applied to compressible signals.

We conclude this section by mentioning the dual formulation of (9)

$$\min_{\boldsymbol{\sigma}} \frac{\tau}{2}\|\boldsymbol{\sigma}\|_2^2 - \boldsymbol{u}^T\boldsymbol{\sigma} \quad \text{subject to} \quad -\mathbf{1} \leq \boldsymbol{\Phi}^*\boldsymbol{\sigma} \leq \mathbf{1}. \tag{14}$$

Although this formulation has not been studied extensively, an active-set method was proposed in [81]. This method solves a sequence of subproblems where a subset of the constraints (corresponding to a subdictionary) is enforced. The dual of each subproblem can each be expressed as a least squares problem over the subdictionary, where the subdictionaries differ by a single column from one problem to the next. The connections between this approach and greedy pursuits are evident. ■

REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[2] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete Fourier information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[4] S. Muthukrishnan, *Data Streams: Algorithms and Applications*. Boston, MA: Now Publishers, 2005.

[5] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2433–2452, Oct. 1998.

[6] R. G. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," 2008, submitted for publication.

[7] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[8] D. Wipf and B. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[9] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian matching pursuit: Model uncertainty and parameter estimation for sparse linear models," *IEEE Trans. Signal Process.*, 2008, submitted for publication.

[10] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 707–710, Oct. 2007.

[11] A. J. Miller, *Subset Selection in Regression*, 2nd ed. London, U.K.: Chapman and Hall, 2002.

[12] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 269–280, Jan. 2010.

[13] D. L. Donoho, A. Maliki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[14] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[15] S. Foucart and M.-J. Lai, "Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$," *Appl. Comput. Harmonic Anal.*, vol. 26, no. 3, pp. 395–407, 2009.

[16] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, pp. 2197–2202, Mar. 2003.

[17] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2845–2862, Nov. 2001.

[18] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Proc. 46th Annu. Allerton Conf. Commun. Control Comput.*, 2008, pp. 798–805.

[19] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and O. Yilmaz, "Sparco: A testing framework for sparse reconstruction," *ACM Trans. Math. Softw.*, vol. 35, no. 4, pp. 1–16, 2009.

[20] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Annu. Asilomar Conf. Signals Syst. Comput.*, Nov. 1993, vol. 1, pp. 40–44.

[21] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximation," *J. Constr. Approx.*, vol. 13, pp. 57–98, 1997.

[22] A. C. Gilbert, M. Muthukrishnan, and M. J. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. 14th Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2003.

[23] J. H. Friedman and W. Stuetzle, "Projection pursuit regressions," *J. Amer Stat. Assoc.*, vol. 76, no. 376, pp. 817–823, Dec. 1981.

[24] R. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Adv. Comput. Math.*, vol. 5, pp. 173–187, 1996.

[25] C. Huang, G. Cheang, and A. R. Barron, "Risk of penalized least-squares, greedy selection, and ℓ_1-penalization for flexible function libraries," *Ann. Stat.*, 2008, submitted for publication.

[26] A. R. Barron, A. Cohen, R. A. DeVore, and W. Dahmen, "Approximation and learning by greedy algorithms," *Ann. Stat.*, vol. 36, no. 1, pp. 64–94, 2008.

[27] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[28] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *J. Signal Process.*, vol. 86, *Special Issue on Sparse Approximations in Signal and Image Processing*, pp. 572–588, Apr. 2006.

[29] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.

[30] T. Zhang, "On the consistency of feature selection using greedy least squares regression," *J. Mach. Learning Res.*, vol. 10, pp. 555–568, 2009.

[31] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[32] A. Fletcher and S. Rangan, "Orthogonal matching pursuit from noisy random measurements: A new analysis," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2009, p. 23.

[33] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit (StOMP)," Stanford Univ., Palo Alto, CA, Stat. Dept. Tech. Rep. 2006-02, Mar. 2006.

[34] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comput. Math.*, vol. 9, no. 3, pp. 317–334, 2009.

[35] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 310–316, Apr. 2009.

[36] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmonic Anal.*, vol. 26, no. 3, pp. 301–321, 2009.

[37] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[38] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," California Inst. Technol., Pasadena, CA, ACM Rep. 2008-01, 2008.

[39] A. Maliki and D. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," Sep. 2009. [Online]. Available: arXiv:0909.0777

[40] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best*k*-term approximation," *J. Amer. Math. Soc.*, vol. 22, no. 1, pp. 211–231, 2009.

[41] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[42] A. Cohen, R. A. DeVore, and W. Dahmen, "Instance-optimal decoding by thresholding in compressed sensing," 2008.

[43] R. Garg and R. Khandekar, "Gradient descent with sparsification: An iterative algorithms for sparse recovery with restricted isometry property," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, Jun. 2009.

[44] R. Meka, P. Jain, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," [Online]. Available: arXiv:0909.5457

[45] J.-L. Starck, M. Elad, and D. L. Donoho, "Redundant multiscale transforms and their application for morphological component analysis," *J. Adv. Imaging Electron Phys.*, vol. 132, pp. 287–348, 2004.

[46] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

[47] M. R. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA J. Numer. Anal.*, vol. 20, pp. 389–403, 2000.

[48] R. Gribonval and M. Nielsen, "Highly sparse representations from dictionaries are unique and independent of the sparseness measure," Aalborg Univ., Aalborg, Denmark, Tech. Rep., Oct. 2003.

[49] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. R. Stat. Soc. B*, vol. 58, pp. 267–288, 1996.

[50] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1341–1344, Jun. 2004.

[51] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.

[52] J. A. Tropp, "On the conditioning of random subdictionaries," *Appl. Comput. Harmonic Anal.*, vol. 25, pp. 1–24, 2008.

[53] E. J. Candès and Y. Plan, "Near-ideal model selection by ℓ_1 minimization," *Ann. Stat.*, vol. 37, no. 5A, pp. 2145–2177, 2009.

[54] J. A. Tropp, "Norms of random submatrices and sparse approximation," *C. R. Acad. Sci. Paris Ser. I Math.*, vol. 346, pp. 1271–1274, 2008.

[55] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2006.

[56] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[57] M. Rudelson and R. Vershynin, "On sparse reconstruction from Fourier and Gaussian measurements," *Commun. Pure Appl. Math.*, vol. 61, no. 8, pp. 1025–1045, 2008.

[58] Y. Zhang, "On the theory of compressive sensing by ℓ_1 minimization: Simple derivations and extensions," Rice Univ., Houston, TX, CAAM Tech. Rep. TR08-11, 2008.

[59] D. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lower dimensions," *J. Amer. Math. Soc.*, vol. 1, pp. 1–53, 2009.

[60] B. Hassibi and W. Xu, "On sharp performance bounds for robust sparse signal recovery," in *Proc. IEEE Symp. Inf. Theory*, Seoul, Korea, 2009, pp. 493–397.

[61] D. L. Donoho and Y. Tsaig, "Fast solution ofℓ_1-norm minimization problems when the solution may be sparse," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[62] M. A. Saunders, "PDCO: Primal-dual interior-point method for convex objectives," Syst. Optim. Lab., Stanford Univ., Stanford, CA, Nov. 2002.

[63] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[64] E. Candès and J. Romberg, "ℓ_1-MAGIC: Recovery of sparse signals via convex programming," California Inst. Technol., Pasadena, CA, Tech. Rep., Oct. 2005.

[65] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[66] I. Daubechies, M. Defriese, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. LVII, pp. 1413–1457, 2004.

[67] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, pp. 1107–1130, 2008.

[68] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 2479–2493, Aug. 2009.

[69] K. Bredies and D. A. Lorenz, "Linear convergence of iterative

soft-thresholding," *SIAM J. Sci. Comput.*, vol. 30, no. 2, pp. 657–683, 2008.

[70] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinking/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[71] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[72] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, "A fast algorithms for sparse reconstruction based on shrinkage, subspace optimization, and continuation," Rice Univ., Houston, TX, CAAM Tech. Rep. 09-01, Jan. 2009.

[73] W. Shi, G. Wahba, S. J. Wright, K. Lee, R. Klein, and B. Klein, "LASSO-Patternsearch algorithm with application to opthalmology data," *Stat. Interface*, vol. 1, pp. 137–153, Jan. 2008.

[74] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, 2008.

[75] Y. Nesterov, "A method for unconstrained convex problem with the rate of convergence $o(1/k^2)$," *Doklady AN SSSR*, vol. 269, pp. 543–547, 1983.

[76] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Norwell, MA: Kluwer, 2004.

[77] A. Nemirovski and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. New York: Wiley, 1983.

[78] Y. Nesterov, "Gradient methods for minimizing composite objective function," Catholic Univ. Louvain, Louvain, Belgium, CORE Discussion Paper 2007/76, Sep. 2007.

[79] A. Beck and M. Teboulle, "A fast iterative shrinkage-threshold algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, pp. 183–202, 2009.

[80] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," Apr. 2009. [Online]. Available: arXiv:0904.3367

[81] M. P. Friedlander and M. A. Saunders, "Active-set approaches to basis pursuit denoising," in *Talk SIAM Optim. Meeting*, Boston, MA, May 2008.

ABOUT THE AUTHORS

Joel A. Tropp (Member, IEEE) received the B.A. degree in Plan II Liberal Arts Honors and the B.S. degree in mathematics from the University of Texas at Austin in 1999. He continued his graduate studies in computational and applied mathematics at UT-Austin where he received the M.S. degree in 2001 and the Ph.D. degree in 2004.

He joined the Mathematics Department, University of Michigan, Ann Arbor, as a Research Assistant Professor in 2004, and he was appointed T. H. Hildebrandt Research Assistant Professor in 2005. Since August 2007, he has been an Assistant Professor of Applied & Computational Mathematics at the California Institute of Technology, Pasadena.

Dr. Tropp's research has been supported by the National Science Foundation (NSF) Graduate Fellowship, the NSF Mathematical Sciences Postdoctoral Research Fellowship, and the Office of Naval Research (ONR) Young Investigator Award. He is also a recipient of the 2008 Presidential Early Career Award for Scientists and Engineers (PECASE), and the 2010 Alfred P. Sloan Research Fellowship.

Stephen J. Wright received the B.Sc. (honors) and Ph.D. degrees from the University of Queensland, Brisbane, Qld., Australia, in 1981 and 1984, respectively.

After holding positions at North Carolina State University, Argonne National Laboratory, and the University of Chicago, he joined the Computer Sciences Department, University of Wisconsin-Madison as a Professor in 2001. His research interests include theory, algorithms, and applications of computational optimization.

Dr. Wright is Chair of the Mathematical Programming Society and serves on the Board of Trustees of the Society for Industrial and Applied Mathematics (SIAM). He has served on the editorial boards of the *SIAM Journal on Optimization*, the *SIAM Journal on Scientific Computing*, *SIAM Review*, and *Mathematical Programming, Series A and B* (the last as Editor-in-Chief).