COMPUTATIONAL METHODS
FOR SPARSE SOLUTION OF LINEAR INVERSE PROBLEMS

JOEL A. TROPP AND STEPHEN J. WRIGHT

Technical Report No. 2009-01
March 2009

APPLIED & COMPUTATIONAL MATHEMATICS
CALIFORNIA INSTITUTE OF TECHNOLOGY
mail code 217-50 · pasadena, ca 91125

Computational Methods for Sparse Solution of Linear Inverse Problems

Joel A. Tropp and Stephen J. Wright

Abstract—**In sparse approximation problems, the goal is to find an approximate representation of a target signal using a linear combination of a few elementary signals drawn from a fixed collection. This paper surveys the major algorithms that are used for solving sparse approximation problems in practice. 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, which makes the algorithms discussed in this paper versatile tools with a wealth of applications.**

Index Terms—**sparse approximation, compressed sensing, matching pursuit, convex optimization**

I. Introduction

LINEAR inverse problems arise throughout engineering and the mathematical sciences. In most applications, these problems are ill-conditioned or underdetermined, so we must apply additional regularizing constraints in order to obtain interesting or useful solutions. The last two decades have witnessed an explosion of interest in regularization via *sparsity constraints*. That is, we seek approximate solutions to linear systems where 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], [3].

Tykhonov regularization, the classical device for solving linear inverse problems, controls the energy of the unknown vector (i.e., the Euclidean norm). This approach leads 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 correctly solve 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 where each method is most effective in practice. Finally, we sketch the theoretical results that justify the application of these algorithms.

Subsection I-A describes "ideal" formulations of sparse approximation and some common features of algorithms for approaching these problems. Section II provides additional detail about greedy pursuit methods. Section III presents formulations based on convex optimization and algorithms for solving these convex programs. Finally, Section IV surveys some new horizons worth exploring.

A. Formulations

Suppose that $\boldsymbol{\Phi} \in \mathbb{R}^{m \times N}$ is a real matrix whose columns have unit Euclidean norm: $\|\boldsymbol{\varphi}_j\|_2 = 1$ for $j = 1, 2, \dots, N$. (The normalization does not compromise generality.) This matrix is often referred to as a *dictionary*. The "entries" in the dictionary are the columns of the matrix, 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 $\|\boldsymbol{x}\|_0 \le s$. When $\boldsymbol{u} = \boldsymbol{\Phi x}$, we refer to $\boldsymbol{x}$ as a *representation* of the signal $\boldsymbol{u}$ with respect to the dictionary.

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 $\varepsilon \ge 0$, in case the observed signal is contaminated with noise:

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

It is most common to measure the prediction–observation discrepancy with the Euclidean norm, but other metrics 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 $s \ge 1$:

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

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

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

JAT is with Applied & Computational Mathematics, Firestone Laboratories MC 217-50, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125-5000 USA. E-mail: jtropp@acm.caltech.edu.

SJW is with the Computer Sciences Department, University of Wisconsin, 1210 W. Dayton St., Madison, WI 53706 USA. E-mail: swright@cs.wisc.edu

JAT was supported by ONR N00014-08-1-2065. SJW was supported by NSF CCF-0430504, DMS-0427689, CTS-0456694, and CNS-0540147.

Date: March 15, 2009.

If there are no restrictions on the dictionary $\mathbf{\Phi}$ 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 last 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 such an important and popular topic of research in recent years.

B. Structured Sparse Models

Sparse approximation has become increasingly important as it has become clear that sparsity constraints are pervasive. Here, we focus on how sparse models affect the efficiency of algorithms. For more details on sparse modeling, see other papers in this volume.

Researchers in mathematical signal processing have demonstrated convincingly that many naturally occurring signals are sparse with respect to dictionaries that can be constructed with methods from harmonic analysis [5]. For example, natural images can be approximated with relatively few wavelet coefficients. As a consequence, in many sparse approximation problems, the dictionary $\mathbf{\Phi}$ has tremendous structure and offers fast matrix–vector multiplications.

In compressive sampling, we typically view $\mathbf{\Phi}$ as the product of a random observation matrix and a fixed orthogonal matrix that determines a basis in which the signal is sparse. For large-scale compressive sampling problems, it is essential that the observation matrix and sparsity basis both admit an efficient matrix–vector multiply, or else algorithms will be hopelessly slow.

C. Major Algorithmic Approaches

There are 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 [6].
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 methods.** 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 [7] or average over most-probable models [8].
4) **Nonconvex optimization.** Relax the ℓ_0 problem to a related nonconvex problem and attempt to identify a stationary point [9].
5) **Brute force.** Search through all possible support sets, possibly using cutting-plane methods to reduce the number of possibilities [10, Sec. 3.7–3.8].

This article focuses on greedy pursuits and convex optimization. These two methods have the advantages that they 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.

D. Verifying Correctness

Researchers have identified several tools which can be used to prove that sparse approximation algorithms produce optimal solutions to sparse approximation problems. These ideas also have an impact on 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 $\mathbf{\Phi}$. Suppose a signal $\boldsymbol{u}$ has two different s-sparse representations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Clearly, we have

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

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

To ensure that sparse approximation is computationally tractable, we need stronger assumptions on $\mathbf{\Phi}$. 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 $\Delta\boldsymbol{u} = \mathbf{\Phi}(\Delta\boldsymbol{x})$. 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 $\mathbf{\Phi}$ satisfies the *restricted isometry property* (RIP) of order K with constant $\delta = \delta_K < 1$ if

$$\|\boldsymbol{x}\|_0 \leq K \;\Rightarrow\; (1-\delta)\|\boldsymbol{x}\|_2^2 \leq \|\mathbf{\Phi}\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2. \quad (4)$$

This concept was introduced in the important paper [11]. For sparse approximation, we hope (4) holds for large K.

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

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

An elementary argument [12] 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 [6], [13].

Certain random matrices, however, satisfy much stronger RIP bounds with high probability. Gaussian matrices, Bernoulli matrices, and random sections of Fourier matrices typically satisfy RIP 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 [11] for discussion.

Recently, researchers have observed that sparse matrices may satisfy a related property, called RIP-1, even when they do not satisfy (4). RIP-1 can also be used to analyze sparse approximation algorithms. See [14] for details.

E. Key Cross-Cutting Issues

Structural properties of the matrix $\mathbf{\Phi}$ have a substantial impact on the implementation of sparse approximation algorithms. In most applications of interest, the large size or lack of sparseness in $\mathbf{\Phi}$ makes it impossible to store this matrix (or any substantial submatrix) explicitly in computer memory. It is often the case, however, that matrix–vector products involving $\mathbf{\Phi}$ and $\mathbf{\Phi}^*$ can be performed efficiently. For example, the cost of these products is $\mathrm{O}(N \log N)$ when $\mathbf{\Phi}$ 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). Nevertheless, all the algorithms discussed below can be implemented in a way that requires access to $\mathbf{\Phi}$ only through matrix–vector products.

Spectral properties of subdictionaries, such as those encapsulated in (4), have additional implications for the computational cost of sparse approximation algorithms. In particular, many methods exhibit fast asymptotic convergence because the RIP ensures that the subdictionaries encountered during execution have superb conditioning. This point is most evident with algorithms that solve least-squares problems iteratively because LSQR and CG are most efficient with well-conditioned matrices. 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.

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 that yield a substantial improvement in approximating the signal. We begin with a description of the simplest effective greedy algorithm, orthogonal matching pursuit (OMP), and its theoretical guarantees. Afterward, we outline a more sophisticated method, called CoSaMP, and its theory. We conclude with some general comments about the role of greedy algorithms in sparse approximation.

A. Orthogonal Matching Pursuit

Orthogonal matching pursuit is one of the earliest methods for sparse approximation. The basic references in the signal processing literature are [15], [16], but the idea can be traced to 1950s work on variable selection in regression [10].

Figure 1 contains a mathematical description of OMP. The symbol $\mathbf{\Phi}_\Omega$ denotes the subdictionary indexed by a subset $\Omega \subset \{1, 2, \dots, N\}$.

Fig. 1. Orthogonal Matching Pursuit (OMP)

```
• Input. A signal $\boldsymbol{u} \in \mathbb{R}^m$, a matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times N}$
• 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 $\mathbf{\Phi}$ that is most strongly correlated with the residual:
       $n_k \in \arg\max_n \; |\langle \boldsymbol{r}_{k-1}, \boldsymbol{\varphi}_n \rangle|$ and
       $\Omega_k = \Omega_{k-1} \cup \{n_k\}$.
3) Estimate. Find the best coefficients for approximating the signal with the columns chosen so far.
       $\boldsymbol{x}_k = \arg\min_{\boldsymbol{y}} \|\boldsymbol{u} - \mathbf{\Phi}_{\Omega_k} \boldsymbol{y}\|_2$.
4) Iterate. Update the residual:
       $\boldsymbol{r}_k = \boldsymbol{u} - \mathbf{\Phi}_{\Omega_k} \boldsymbol{x}_k$.
   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.
```

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 $\mathbf{\Phi}^* \boldsymbol{r}_{k-1}$, 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 [17]. In certain applications, such as ridge regression, the "columns" of $\mathbf{\Phi}$ are indexed by a continuous parameter, and identification can be posed as a low-dimensional optimization problem.

The estimation step requires the solution of a least-squares problem. The most common technique is to maintain a QR factorization of $\mathbf{\Phi}_{\Omega_k}$, 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: $\|\boldsymbol{r}_k\|_2 \le \varepsilon$.
- Halt when no column explains a significant amount of energy in the residual: $\|\mathbf{\Phi}^* \boldsymbol{r}_{k-1}\|_\infty \le \varepsilon$.

These criteria can all be implemented at minimal cost.

As with other algorithms, OMP can take advantage of a dictionary that offers a fast matrix–vector product; see Section I-E.

A huge number of related greedy pursuit algorithms have been proposed in the literature; we cannot do them justice here. Some noteworthy variants include matching pursuit [6], the relaxed greedy algorithm [18], and the ℓ_1-penalized greedy algorithm [19].

B. Guarantees

OMP produces the residual $\boldsymbol{r}_m = \mathbf{0}$ 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 [6]. For example, OMP offers the estimate

$$\begin{aligned}&\|\boldsymbol{r}_k\|_2 \le (1-\varrho^2)^{k/2}\,\|\boldsymbol{u}\|_2, \quad \text{where}\\ &\varrho = \inf_{\|\boldsymbol{v}\|_2=1}\sup_n |\langle \boldsymbol{v}, \boldsymbol{\varphi}_n\rangle|.\end{aligned}$$

See [16, Sec. 3] for details. Unfortunately, the covering parameter ϱ is $\mathrm{O}(m^{-1/2})$ unless $N = \mathrm{O}(\mathrm{e}^m)$. Since ρ is typically quite small, this type of result 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 [18, Eqn. (1.9)]. For example, OMP offers the estimate

$$\begin{aligned}&\|\boldsymbol{r}_k\|_2 \le k^{-1/2}\,\|\boldsymbol{u}\|_{\boldsymbol{\Phi}}, \quad \text{where}\\ &\|\boldsymbol{u}\|_{\boldsymbol{\Phi}} = \inf\{\|\boldsymbol{x}\|_1 : \boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}\}.\end{aligned}$$

The dictionary seminorm $\|\cdot\|_{\boldsymbol{\Phi}}$ is typically small when its argument has a good sparse approximation. For further improvements on this estimate, see [20]. This bound is usually superior to the exponential rate estimate, 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 [17], [21]. For example, if $3\mu k \le 1$, then

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

where $\boldsymbol{a}_k^\star$ denotes the best ℓ_2 approximation of $\boldsymbol{u}$ as a linear combination of k columns from $\boldsymbol{\Phi}$. See [22], [23], [24] for refinements.

Finally, we remark that, if $\boldsymbol{\Phi}$ is sufficiently random, then OMP provably recovers sparse signals. See [25].

C. CoSaMP

Research on greedy pursuits has recently culminated in a new algorithm, called CoSaMP, that offers optimal performance guarantees. The method was designed for compressive sampling, but it also offers attractive guarantees for classical sparse approximation problems. CoSaMP economizes on matrix–vector multiplications, so it is most valuable when these products can be computed efficiently [26]. A related algorithm appears in [27].

Figure 2 contains a description of CoSaMP. The symbol $[\boldsymbol{x}]_r$ denotes the restriction of $\boldsymbol{x}$ to the r components largest in magnitude, ties broken lexicographically.

To ensure that CoSaMP behaves well, we assume throughout this section that the dictionary $\boldsymbol{\Phi}$ verifies the RIP (4) of order $2s$ with constant $\delta_{2s} \ll 1$. As noted, this hypothesis holds for reasonable values of s when $\boldsymbol{\Phi}$ is incoherent or suitably random. Of course, the algorithm can be applied without the RIP, but its performance may be unpredictable.

In the implementation, it is important to use an iterative algorithm to solve the least-squares problem in the estimation step. The RIP ensures that these subproblems are always well conditioned, so iterative methods converge quickly. As a consequence, each outer iteration of the algorithm requires at most a constant number of matrix–vector multiplications, and this cost dominates the computation. See Section I-E for further discussion of this point.

Fig. 2. Compressive Sampling Matching Pursuit (CoSaMP) variant

```
• Input. A signal $u \in \mathbb{R}^m$, a matrix $\Phi \in \mathbb{R}^{m \times N}$, sparsity parameter $s$
• Output. An $s$-sparse coefficient vector $x \in \mathbb{R}^N$

1) Initialize. Set the initial coefficient vector $x_0 = 0$ and the residual $r_0 = u$. Let $k = 1$.
2) Identify. Find $2s$ columns of $\Phi$ that are most strongly correlated with the residual:
       $\Omega \in \arg\min_{|T| \le 2s} \sum_{n \in T} |\langle r_{k-1}, \varphi_n \rangle|.$
3) Estimate. Find the best coefficients for approximating the residual with the chosen columns:
       $y_k = \arg\min_{y} \|r_{k-1} - \Phi_\Omega y\|_2$
4) Prune. Combine the old and new coefficient vectors and retain the $s$ largest components:
       $z = x_{k-1} + y_k$   and   $x_k = [z]_s.$
5) Iterate. Update the residual:
       $r_k = u - \Phi x_k.$
   Repeat (2)–(5) until stopping criterion holds.
6) Output. Return $x = x_k$.
```

Needell and Tropp demonstrate that each iteration of the CoSaMP algorithm reduces the approximation error by a constant factor until it approaches its minimal value. To be specific, suppose the signal $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}^\star + \boldsymbol{e}$ for arbitrary 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

$$\|\boldsymbol{x}^\star - \boldsymbol{x}\|_2 \le Cs^{-1/2}\|\boldsymbol{x}^\star - [\boldsymbol{x}^\star]_{s/2}\|_1 + C\|\boldsymbol{e}\|_2, \qquad (5)$$

where C is a constant. No algorithm can produce an essentially smaller error for general input signals.

This error bound allows us to develop stopping criteria for CoSaMP that are tailored to the signals of interest. For example, when the sorted entries of the coefficient vector $\boldsymbol{x}^\star$ decay polynomially, it can be verified that the algorithm requires only $\mathrm{O}(\log N)$ iterations.

It is worth noting a strong similarity between CoSaMP and iterative thresholding algorithms. See [28] for some related results.

D. Commentary

Greedy pursuit methods have often been considered naïve, in part because there are contrived examples where the greedy approach fails spectacularly. (See [1, Sec. 2.3.2] for an exposition of this claim.) Recent research has vindicated greedy pursuits by demonstrating that they succeed in many of the situations where convex relaxation works. Still, it is misleading to think of greedy methods and convex relaxation methods as distinct approaches to sparse approximation. Indeed, the greedy selection technique is closely related to dual coordinate ascent algorithms (see Section III-F). Similarly, certain methods for convex relaxation, such as LARS [29] and homotopy [30], use a type of greedy selection at each iteration, which is based on the desire to solve an underlying parametrized optimization problem.

Greedy pursuits and related methods (such as homotopy) are sometimes quite fast, especially in the ultrasparse regime in which the number of nonzeros in the representation is very small. Greedy techniques can sometimes be applied to situations in which convex relaxation cannot be used. For example, when the dictionary contains a continuum of elements, the greedy approach can reduce sparse approximation to a sequence of simple one-dimensional optimization problems.

Another advantage of greedy methods is that they can incorporate constraints that do not fit naturally into convex programming formulations. For example, the data stream community has proposed efficient 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 [31]. This is an exciting and important line of work.

III. Optimization

Another fundamental approach to sparse approximation replaces the combinatorial ℓ_0 function in the mathematical programs from Subsection I-A with the ℓ_1 norm, which yields *convex* optimization problems that admit tractable algorithms. In a concrete sense [32], the ℓ_1 norm is the closest convex function to the ℓ_0 function, so this relaxation is very natural.

The convex relaxation of the equality-constrained problem becomes

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{u}, \tag{6}$$

and the mixed formulation becomes

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{u}\|_2^2 + \tau\|\boldsymbol{x}\|_1. \tag{7}$$

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 (7) repeatedly for different choices of this parameter, or even to trace the complete path of solutions as τ decreases toward zero. When $\tau \geq \|\boldsymbol{\Phi}^*\boldsymbol{u}\|_\infty$, the solution of (7) is $\boldsymbol{x} = \boldsymbol{0}$.

Another variant is the LASSO formulation [33], which first arose in the context of variable selection:

$$\min_{\boldsymbol{x}} \|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{u}\|_2^2 \;\; \text{subject to} \;\; \|\boldsymbol{x}\|_1 \leq \beta. \tag{8}$$

The LASSO is equivalent to (7) in the sense the path of solutions to (8) parameterized by positive β matches the solution path for (7) as τ varies. Finally, we note another common formulation

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \;\; \text{subject to} \;\; \|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{u}\|_2 \leq \varepsilon \tag{9}$$

that explicitly parameterizes the error constraint.

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 [13], [12], [21] establish that the equality-constrained problem (6) 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 [34], [35], [23] showed that the convex programs (7) and (9) 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 [36] proves that that the equality-constrained problem (6) can identify random signals, even when the sparsity level s has order $m/\log m$. Most recently, the paper [37] observed that ideas from [35], [38] imply that the convex relaxation (7) can identify noisy, random sparse signals in a similar parameter regime.

Results from [11], [39] demonstrate that convex relaxation succeeds well in the presence of the RIP. For example, suppose that the signal $\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{x}^\star + \boldsymbol{e}$ and the dictionary has RIP constant $\delta_{2s} \ll 1$. Then the solution $\boldsymbol{x}$ to (9) verifies

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

provided that $\varepsilon \geq \|\boldsymbol{e}\|_2$. Compare this bound with the result (5) for CoSaMP.

Finally, let us mention an alternative approach for analyzing convex relaxation algorithms that relies on geometric properties of the kernel of the dictionary [40], [41], [42].

B. Active Set / Pivoting

Pivoting algorithms explicitly trace the path of solutions as the scalar parameter in (8) ranges over a set of values. This approach relies on the fact that the solution to (8) is a piecewise-linear function of β, a consequence of the fact that the optimality (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 closest value of β where the derivative of the piecewise-linear function changes.

The homotopy method of Osborne, Presnell, and Turlach [30] follows this approach. The algorithm starts with $\beta = 0$, where the solution of (8) is identically zero, and it progressively locates the next largest value of β where a column enters or exits the active set. The efficiency of the algorithm depends on updating and downdating a QR factorization of the active columns. A similar method [29] is implemented as `SolveLasso` in the SparseLab toolbox[1]. Similar approaches can be developed for (7).

If we limit our attention to values of β where the number of nonzero entries in $\boldsymbol{x}$ remains small, the active set approach

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

is reasonably efficient. If we apply the homotopy method until s nonzero components are identified, the workload consists of approximately $2s$ matrix–vector multiplications by $\mathbf{\Phi}$ or $\mathbf{\Phi}^*$, together with $O(ms^2)$ operations for updating the factorization and performing other linear algebra operations. This cost is comparable with OMP.

Indeed, OMP and homotopy are quite similar. In each case, the set of nonzero components is refined by progressively adding components and updating the solution of a reduced linear least-squares problem. The criterion for selecting components involves the inner products between inactive columns of $\mathbf{\Phi}$ and the current residual $\boldsymbol{u}-\mathbf{\Phi}\boldsymbol{x}$. One notable difference is that homotopy occasionally rejects columns that have already been chosen. See [29], [43] for other comparisons.

Fig. 3. Sparse Reconstruction via Separable Approximation (SpaRSA)

```
• 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 $\alpha_k$ and obtain $x_k^+$ from (10a). If an acceptance test on $x_k^+$ is not passed, increase $\alpha_k$ by some factor and repeat.
3) Line Search. Choose $\gamma_k \in (0, 1]$ and obtain $x_{k+1}$ from (10b).
4) Test. If stopping criterion holds, terminate with $x = x_{k+1}$. Otherwise, set $k \leftarrow k + 1$ and go to (2).
```

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], [44] 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. A crucial aspect of this approach is that the algorithms take advantage of fast matrix–vector multiplications. An implementation is available as pdco and SolveBP in the SparseLab toolbox.

Other interior-point methods have also been proposed expressly for compressive sampling problems. The paper [45] describes a primal log-barrier approach for a quadratic reformulation of (7):

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

The technique relies on a specialized preconditioner that allows the internal Newton iterations to be completed efficiently with CG. The method[2] is available as l1_ls. The ℓ_1-magic package[3] [46] contains a primal log-barrier code for the second-order cone formulation (9), 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 Subsection III-D on problems with very sparse solutions, and they do not benefit from warm starts to the same extent as the other approaches in this section. On the other hand, their performance is relatively insensitive to the sparsity of the solution or the value of the regularization parameter. Interior-point methods appear to be more robust in the sense that there are few cases of very slow performance or outright failure, which sometimes occurs with other approaches.

D. Gradient Methods

Gradient descents, also known as *first-order* methods, are iterative algorithms for solving (7) in which the major operation at each iteration is to compute gradient of the least-squares term at the current iterate, viz., $\mathbf{\Phi}^*(\mathbf{\Phi}\boldsymbol{x}_k-\boldsymbol{u})$. Many of these methods compute the next iterative $\boldsymbol{x}_{k+1}$ using the rules

$$\begin{aligned}\boldsymbol{x}_k^+ &:= \arg\min_{\boldsymbol{z}} (\boldsymbol{z}-\boldsymbol{x}_k)^*\mathbf{\Phi}^*(\mathbf{\Phi}\boldsymbol{x}_k-\boldsymbol{u}) \\ &\qquad + \frac{1}{2}\alpha_k\|\boldsymbol{z}-\boldsymbol{x}_k\|_2^2 + \tau\|\boldsymbol{z}\|_1, \qquad (10a) \\ \boldsymbol{x}_{k+1} &:= \boldsymbol{x}_k + \gamma_k(\boldsymbol{x}_k^+ - \boldsymbol{x}_k). \qquad (10b)\end{aligned}$$

for some choice of scalar parameters α_k and γ_k. Alternatively, we can write the subproblem (10a) as

$$\begin{aligned}\boldsymbol{x}_k^+ := \arg\min_{\boldsymbol{z}} \frac{1}{2}\left\|\boldsymbol{z}-\left(\boldsymbol{x}_k - \frac{1}{\alpha_k}\mathbf{\Phi}^*(\mathbf{\Phi}\boldsymbol{x}_k-\boldsymbol{u})\right)\right\|_2^2 \\ + \frac{\tau}{\alpha_k}\|\boldsymbol{z}\|_1. \qquad (11)\end{aligned}$$

Figure 3 specifies a representative algorithm, called SpaRSA, that falls into the *operator-splitting* framework of [47]. Other aliases include *iterative splitting and thresholding (IST)* [48] and *fixed-point iteration* [49].

"Standard" convergence results for these methods, e.g., [47, Theorem 3.4], require that $\sup_k \alpha_k > \|\mathbf{\Phi}^*\mathbf{\Phi}\|_2/2$, a tight restriction that usually leads to slow convergence in practice. The more practical variants described in [50] admit smaller values of α_k, provided that a sufficient decrease in the objective in (7) occurs over a span of successive iterations. One particular approach uses a Barzilai–Borwein formula that chooses a value of α_k in the spectrum of $\mathbf{\Phi}^*\mathbf{\Phi}$. When $\boldsymbol{x}_k^+$ fails the acceptance test in Step 2, the parameter α_k is increased (repeatedly, as necessary) by a constant factor. Steplengths $\gamma_k \equiv 1$ are used in [49] and [50].

Several variants of the SpaRSA have appeared. The "iterated hard shrinkage" method of [51] sets $\alpha_k \equiv 0$ in (10) and chooses γ_k to do a conditional minimization along the search direction. TwIST [52], a variant of IST that is significantly faster in practice, deviates from the SpaRSA framework in that the previous iterate $\boldsymbol{x}_{k-1}$ also enters into the step calculation, in the manner of successive over-relaxation approaches in other areas of scientific computation. GPSR [53] is another approach for solving (7) that can be viewed as a gradient-projection algorithm for the convex quadratic program obtained by splitting $\boldsymbol{x}$ into positive and negative parts.

[2] www.stanford.edu/∼boyd/l1_ls/

[3] www.l1-magic.org

The SpaRSA approach works well on sparse signals when $\mathbf{\Phi}$ satisfies the RIP. For these problems, it tends to identify the nonzero components of $\boldsymbol{x}$ quickly, after which the method behaves essentially like an iterative least-squares method. Because of the RIP, these final iterates converge quickly. Computationally, these steps are quite similar to the estimation step of CoSaMP.

When the solution of (7) is not particularly sparse or the regularization parameter τ is small, gradient approaches may be ineffective, converging slowly or not at all. In this setting, *warm starts* can vastly improve the behavior of these approaches. That is, we can dramatically reduce the number of iterations when the initial estimate $\boldsymbol{x}_0$ in Step 1 is close to the true solution. This observation motivates the design of *continuation* strategies, in which we solve (7) for a decreasing sequence of values of τ, using the approximate solution for each value as the starting point for the next subproblem. It is evident that continuation is related to pivoting strategies of Subsection III-B that track individual changes in the active components of $\boldsymbol{x}$ explicitly.

Some explicit continuation methods are described in [49], [50]. Though adaptive strategies for choosing the sequence of τ values have been proposed, the design of a robust, practical, and theoretically effective continuation algorithm remains an important 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 [53], [50], this enhancement is made only after the first-order algorithm is terminated as a means of removing the bias in the formulation (7) introduced by the regularization term. Other methods [54] use second-order information at intermediate steps of the algorithm, much like two-metric gradient projection [55]. (A similar approach was proposed for the related problem of ℓ_1-regularized logistic regression in [56].) To compute the second-order components of the steps, iterative methods can be used to find approximate solutions to unconstrained least-squares subproblems involving only the nonzero components of $\boldsymbol{x}$. These subproblems are, of course, closely related to the ones that arise in the matching pursuit algorithms of Section II.

A different type of gradient projection approach is described by [57, Section 4], which considers the formulation (8). This approach takes steps along the negative gradient of the least-squares objective in (8), with steplength chosen by a Barzilai–Borwein formula (with backtracking to ensure monotonicity) and projects the resulting vector onto the constraint set $\|\boldsymbol{x}\|_1 \leq \beta$. The ultimate goal in [57] is to solve (9) for a given value of ε. Then a scalar equation is solved to identify the value of β for which the solution of (8) coincides with the solution of (9) for the given ε.

Finally, we mention that optimal gradient methods for convex minimization [58], [59], [60] can be applied to solve the formulation (7). 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. (In contrast, standard methods 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 solve (7) using optimal gradient methods include the papers [61], [62].

The favorable asymptotic properties of optimal first-order methods do not manifest themselves for problems with relatively large values of τ or with very sparse solutions. These algorithms can be competitive with other gradient methods when τ is small, although the poor performance of methods such as SpaRSA on such problems can be improved significantly by the use of continuation.

Recently, Nemirovski and coworkers have proposed a technique for solving the formulation (9) by means of robust optimal gradient algorithms for stochastic optimization [63]. This approach interprets matrix–vector products as expectations of a random variable. The method appears to be very effective when the dictionary lacks a fast multiply.

F. Dual-Based Algorithms

Greedy pursuit methods are strongly connected with a dual formulation of the problem (7):

$$\min_{\boldsymbol{\sigma}} \frac{\tau}{2}\|\boldsymbol{\sigma}\|_2^2 - \boldsymbol{u}^T\boldsymbol{\sigma} \quad \text{subject to} \quad -\mathbf{1} \leq \mathbf{\Phi}^*\boldsymbol{\sigma} \leq \mathbf{1}. \qquad (12)$$

An active-set method for this formulation [64] solves a sequence of subproblems where a subset of the constraints (corresponding to a subdictionary) is enforced. By converting back to the primal, these subproblems can each be expressed as a least-squares problem over this subdictionary. Typically, the subdictionaries differ by a single column from one problem to the next. These approaches have not been explored extensively in the optimization literature.

IV. Horizons

Test problem collections representative of sparse approximation problems encountered in practice are crucial to guiding further development of sparse reconstruction algorithms. Most algorithmic papers report results only on synthetic test cases. The most significant effort in this direction to date is Sparco [65], a Matlab environment for interfacing algorithms and constructing test problems that also includes a variety of problems gathered from the literature.

Many of the algorithms we describe in this paper lend themselves well to implementation on commodity graphical processing units (GPUs), for certain matrices $\mathbf{\Phi}$. This inexpensive hardware allows remarkable increases in speed over conventional CPU implementations, for certain kinds of compute-intensive, multithreaded numerical computations. For sensing matrices consisting of randomly selected rows of the discrete cosine transformation, an implementation of the SpaRSA strategy is described in [66]. Availability of a GPU implementation of the FFT is crucial to the success of this approach. Multicore and GPU implementations of first-order methods are described also in [67].

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. Info. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

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

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

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

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

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

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

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

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

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

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

[13] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2845–2862, Nov. 2001.

[14] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," 2008, submitted for publication.

[15] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. of the 27th Annual Asilomar Conference on Signals, Systems and Computers*, Nov. 1993.

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

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

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

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

[20] 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.

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

[22] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *J. Signal Process.*, vol. 86, pp. 572–588, April 2006, special issue, "Sparse approximations in signal and image processing".

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

[24] T. Zhang, "On the consistency of feature selection using greedy least squares regression," *J. Mach. Learning Res.*, 2008, to appear.

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

[26] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Anal.*, 2008, to appear.

[27] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," Mar. 2008, submitted for publication.

[28] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," 2008, submitted to *Appl. Comp. Harmonic Anal.*

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

[30] M. R. Osborne, B. Presnell, and B. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, pp. 389–403, 2000.

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

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

[33] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society B*, vol. 58, pp. 267–288, 1996.

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

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

[36] ——, "On the conditioning of random subdictionaries," *Appl. Comp. Harmonic Anal.*, vol. 25, pp. 1–24, 2008.

[37] E. J. Candès and Y. Plan, "Near-ideal model selection by ℓ_1 minimization," Dec. 2007, submitted for publication.

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

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

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

[41] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," 2006, submitted for publication.

[42] Y. Zhang, "On the theory of compressive sensing by ℓ_1 minimization: Simple derivations and extensions," Rice Univ., CAAM Technical Report TR08-11, 2008.

[43] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," Department of Statistics, Stanford University, Preprint, 2006.

[44] M. A. Saunders, "PDCO: primal-dual interior-point method for convex objectives," Systems Optimization Laboratory, Stanford University, Tech. Rep., November 2002.

[45] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A method for large-scale ℓ_1-regularized least squares problems with applications in signal processing and statistics," Electrical Engineering Department, Stanford University, Technical Report, February 2007.

[46] E. Candès and J. Romberg, "ℓ_1-MAGIC: Recovery of sparse signals via convex programming," California Institute of Technology, Tech. Rep., October 2005.

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

[48] I. Daubechies, M. Defriese, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications in Pure and Applied Mathematics*, vol. LVII, pp. 1413–1457, 2004.

[49] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing," CAAM, Rice University, CAAM Technical Report TR07-07, May 2007.

[50] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," University of Wisconsin-Madison, Computer Sciences Department, Technical Report, August 2008, to appear in *IEEE Transactions on Signal Processing*.

[51] K. Bredies and D. A. Lorenz, "Linear convergence of iterative soft-thresholding," *SIAM Journal on Scientific Computing*, vol. 30, no. 2, pp. 657–683, 2008.

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

[53] 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 Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, December 2007.

[54] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, “A fast algorithms for sparse reconstruction based on shrinkage, subspace optimization, and continuation,” Department of Computational and Applied Mathematics, Rice University, CAAM Technical Report 09-01, January 2009.

[55] D. P. Bertsekas, “Projected Newton methods for optimization problems with simple constraints,” *SIAM Journal on Control*, vol. 20, pp. 221–246, 1982.

[56] W. Shi, G. Wahba, S. J. Wright, K. Lee, R. Klein, and B. Klein, “LASSO-Patternsearch algorithm with application to opthalmology data,” *Statistics and its Interface*, vol. 1, pp. 137–153, January 2008.

[57] E. van den Berg and M. P. Friedlander, “Probing the Pareto frontier for basis pursuit solutions,” *SIAM Journal of Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[58] 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.

[59] ——, *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[60] A. Nemirovski and D. B. Yudin, *Problem complexity and method efficiency in optimization*. John Wiley, 1983.

[61] Y. Nesterov, “Gradient methods for minimizing composite objective function,” CORE, Catholic University of Louvain, CORE Discussion Paper 2007/76, September 2007.

[62] A. Beck and M. Teboulle, “A fast iterative shrinkage-threshold algorithm for linear inverse problems,” Technion-Israel Institute of Technology, Technical Report, July 2008.

[63] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[64] M. P. Friedlander and M. A. Saunders, “Active-set approaches to basis pursuit denoising,” May 2008, talk at SIAM Optimization Meeting.

[65] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and O. Yilmaz, “Sparco: A testing framework for sparse reconstruction,” Department of Computer Science, University of British Columbia, Technical Report TR-2007-20, October 2007, revised July, 2008.

[66] S. Lee and S. J. Wright, “Implementing algorithms for signal and image reconstruction on graphical processing units,” Computer Sciences Department, University of Wisconsin-Madison, Tech. Rep., November 2008.

[67] A. Borghi, J. Darbon, S. Peyronnet, T. F. Chan, and S. Osher, “A simple compressive sensing algorithm for parallel many-core architectures,” Department of Mathematics, UCLA, CAM Report 08-64, September 2008.