Sparse Reconstruction by Separable Approximation

Stephen J. Wright, Robert D. Nowak, *Senior Member, IEEE*,
Mário A. T. Figueiredo, *Senior Member, IEEE*

***Abstract*—Finding sparse approximate solutions to large underdetermined linear systems of equations is a common problem in signal/image processing and statistics. Basis pursuit, the least absolute shrinkage and selection operator (LASSO), wavelet-based deconvolution and reconstruction, and compressed sensing (CS) are a few well-known areas in which problems of this type appear. One standard approach is to minimize an objective function that includes a quadratic (ℓ_2) error term added to a sparsity-inducing (usually ℓ_1) regularizater. We present an algorithmic framework for the more general problem of minimizing the sum of a smooth convex function and a nonsmooth, possibly nonconvex regularizer. We propose iterative methods in which each step is obtained by solving an optimization subproblem involving a quadratic term with diagonal Hessian (*i.e.*, separable in the unknowns) plus the original sparsity-inducing regularizer; our approach is suitable for cases in which this subproblem can be solved much more rapidly than the original problem. Under mild conditions (namely convexity of the regularizer), we prove convergence of the proposed iterative algorithm to a minimum of the objective function.**

In addition to solving the standard $\ell_2-\ell_1$ case, our framework yields efficient solution techniques for other regularizers, such as an ℓ_∞ norm and group-separable regularizers. It also generalizes immediately to the case in which the data is complex rather than real. Experiments with CS problems show that our approach is competitive with the fastest known methods for the standard $\ell_2-\ell_1$ problem, as well as being efficient on problems with other separable regularization terms.

***Index Terms*—Sparse Approximation, Compressed Sensing, Optimization, Reconstruction.**

I. Introduction

A. Problem Formulation

In this paper we propose an approach for solving unconstrained optimization problems of the form

$$\min_{\mathbf{x}} \quad \phi(\mathbf{x}) := f(\mathbf{x}) + \tau\, c(\mathbf{x}), \tag{1}$$

where $f:\mathbb{R}^n \to \mathbb{R}$ is a smooth function, and $c:\mathbb{R}^n \to \mathbb{R}$, usually called the *regularizer* or *regularization function*, is finite for all $\mathbf{x}\in\mathbb{R}^n$, but usually nonsmooth and possibly also nonconvex. Problem (1) generalizes the now famous $\ell_2-\ell_1$ problem (called *basis pursuit denoising* (BPDN) in [15])

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1, \tag{2}$$

S. Wright is with Department of Computer Sciences, University of Wisconsin, Madison, WI 53706, **USA**. R. Nowak is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706, **USA**. M. Figueiredo is with the *Instituto de Telecomunicações* and Department of Electrical and Computer Engineering, *Instituto Superior Técnico,* 1049-001 Lisboa, **Portugal**.

This work was partially supported by NSF Grants DMS-0427689, CCF-0430504, CTS-0456694, CNS-0540147, NIH Grant R21EB005473, DOE Grant DE-FG02-04ER25627, and by *Fundação para a Ciência e Tecnologia*, POSC/FEDER, grant POSC/EEA-CPS/61271/2004.

where $\mathbf{y}\in\mathbb{R}^k$, $\mathbf{A}\in\mathbb{R}^{k\times n}$ (usually $k<n$), $\tau\in\mathbb{R}^+$, $\|\cdot\|_2$ denotes the standard Euclidean norm, and $\|\cdot\|_p$ stands for the ℓ_p norm (for $p\geq 1$), defined as $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$. Problem (2) is closely related to the following two formulations:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq T, \tag{3}$$

frequently referred to as the *least absolute shrinkage and selection operator* (LASSO) [70], and

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \leq \varepsilon, \tag{4}$$

where ε and T are nonnegative real parameters. These formulations can all be used to identify sparse approximate solutions to the underdetermined system $\mathbf{y}=\mathbf{A}\mathbf{x}$, and have become familiar in the past few decades, particularly in statistics and signal/image processing contexts. A large amount of research has been aimed at finding fast algorithms for solving these formulations; early references include [16], [55], [66], [69]. For brief historical accounts on the use of the ℓ_1 penalty in statistics and signal processing, see [59] and [71]. The precise relationship between (2), (3), and (4) is discussed in [39] and [75], for example.

Problems with form (1) arise in wavelet-based image/signal reconstruction and restoration (namely deconvolution) [34], [36], [37]. In these problems, $f(\mathbf{x}) = \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2/2$ (as in (2)), with matrix $\mathbf{A}$ having the form $\mathbf{A}=\mathbf{R}\mathbf{W}$, where $\mathbf{R}$ is (the matrix representing) the observation operator (*e.g.*, a convolution with a blur kernel or a tomographic projection); $\mathbf{W}$ contains a wavelet basis or redundant dictionary (*i.e.*, multiplying by $\mathbf{W}$ corresponds to performing an inverse wavelet transform); and $\mathbf{x}$ is the vector of representation coefficients of the unknown image/signal. In wavelet-based image restoration, the regularizer c is often the p-th power of an ℓ_p norm, resulting from adopting generalized Gaussian priors for the wavelet coefficients of natural images [60], although other regularizers have been considered (*e.g.*, [35], [43], [44]).

A popular new application for the optimization problems above is *compressive sensing*[1] (CS) [9], [10], [27]. Recent results show that a relatively small number of random projections of a sparse signal can contain most of its salient information. In the noiseless setting, accurate approximations can be obtained by finding a sparse signal that matches the random projections of the original signal, a problem which can be cast as (4). Problem (2) is a robust version of this reconstruction process, which is resilient to errors and noisy data; this and similar criteria have been proposed and analyzed in [11], [52], [81].

[1] A comprehensive, and frequently updated repository of CS literature and software can be found in `www.dsp.ece.rice.edu/cs/`.

B. Overview of the Proposed Approach

Our approach to solving problems of the form (1) works by generating a sequence of iterates $\{\mathbf{x}^t,\ t = 0, 1, \ldots\}$ and is tailored to problems in which the following subproblem can be set up and solved efficiently at each iteration:

$$\mathbf{x}^{t+1} \in \arg\min_{\mathbf{z}}\ (\mathbf{z} - \mathbf{x}^t)^T \nabla f(\mathbf{x}^t) + \frac{\alpha_t}{2} \|\mathbf{z} - \mathbf{x}^t\|_2^2 + \tau\, c(\mathbf{z}), \tag{5}$$

for some $\alpha_t \in \mathbb{R}^+$. More precisely, we mean that it is much less expensive to compute the gradient ∇f and to solve (5) than it is to solve the original problem (1) by other means. An equivalent form of subproblem (5) is

$$\mathbf{x}^{t+1} \in \arg\min_{\mathbf{z}}\ \frac{1}{2} \left\|\mathbf{z} - \mathbf{u}^t\right\|_2^2 + \frac{\tau}{\alpha_t}\, c(\mathbf{z}), \tag{6}$$

where

$$\mathbf{u}^t = \mathbf{x}^t - \frac{1}{\alpha_t} \nabla f(\mathbf{x}^t). \tag{7}$$

This form is considered frequently in the literature, often under the name of iterative shrinkage/thresholding (IST) algorithms, discussed below. The proximity operator in Combettes and Wajs [17, equation (2.13)] has the form of (6), and is central to the algorithms studied in that paper, which are also suitable for situations in which (5) can be solved efficiently.

Many choices of objective function f and regularizer c in (1) satisfy the assumptions in the previous paragraph. A particularly important case is the one in which c is *separable* into the sum of functions of the individual components of its argument, that is,

$$c(\mathbf{x}) = \sum_{i=1}^{n} c_i(x_i). \tag{8}$$

The ℓ_1 regularizer in (2) obviously has this form (with $c_i(z) = |z|$), as does the ℓ_p^p regularization function $c(\mathbf{z}) = \|\mathbf{z}\|_p^p = \sum_i |z_i|^p$. Also of interest are *group separable* (GS) regularizers, which have the form

$$c(\mathbf{x}) = \sum_{i=1}^{m} c_i(\mathbf{x}_{[i]}), \tag{9}$$

where $\mathbf{x}_{[1]}, \mathbf{x}_{[2]}, \ldots, \mathbf{x}_{[m]}$ are m disjoint subvectors of $\mathbf{x}$. Such regularizers are suitable when there is a natural group structure in $\mathbf{x}$, which is the case, *e.g.*, in the following applications:

- In brain imaging, the voxels associated with different functional regions (for example, motor or visual cortices) may be grouped together in order to identify a sparse set of regional events. In [5], [6], [7] a novel IST algorithm[2] was proposed for solving GS-ℓ_2 (*i.e.*, where $c_i(\mathbf{w}) = \|\mathbf{w}\|_2$) and GS-$\ell_\infty$ (*i.e.*, where $c_i(\mathbf{w}) = \|\mathbf{w}\|_\infty = \max\{|w_i|\}$) problems.
- A GS-ℓ_2 penalty was proposed for source localization in sensor arrays [57]; second-order cone programming was used to solve the optimization problem.
- In gene expression analysis, some genes are organized in functional groups. This has motivated an approach called *composite absolute penalty* [79], which has the form (9), and uses a greedy optimization scheme [80].
- GS regularizers have also been proposed for ANOVA regression [54], [58], [78], and Newton-type optimization methods have been proposed in that context. An interior-point method for the GS-ℓ_∞ case was described in [74].

Another interesting type of regularizer is the total-variation (TV) norm [64], which is of particular interest for image restoration problems [13]. This function is not separable in the sense of (8) or (9), though it is the sum of terms that each involve only a few components of $\mathbf{x}$. The subproblem (5) has the form of an *image denoising* problem, for which efficient algorithms are known (see, for example, [12], [20], [38], [45]).

In the special case of $c(\mathbf{x}) \equiv 0$, the solution of (5) is simply

$$\mathbf{x}^{t+1} = \mathbf{u}^t = \mathbf{x}^t - \frac{1}{\alpha_t} \nabla f(\mathbf{x}^t),$$

so the method reduces to steepest descent on f with adjustment of the step length (line search) parameter.

For the first term f in (1), we are especially interested in the sum-of-squares function $f(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$, as in (2). If the matrix $\mathbf{A}$ is too large (and too dense) to be handled explicitly, it may still be possible to compute matrix-vector products involving $\mathbf{A}$ or its transpose efficiently. If so, computation of ∇f and implementation of the approach described here may be carried out efficiently. We emphasize, however, that the approach we describe in this paper can be applied to any smooth function f.

Observe that the first two terms in the objective function in (5), that is, $(\mathbf{z} - \mathbf{x}^t)^T \nabla f(\mathbf{x}^t) + \frac{\alpha_t}{2} \|\mathbf{z} - \mathbf{x}^t\|_2^2$, can be viewed as a quadratic separable approximation to f about $\mathbf{x}^t$ (up to a constant), that interpolates the first-derivative information and uses a simple diagonal Hessian approximation $\alpha_t \mathbf{I}$ to the second-order term. For this reason, we refer to the approach presented in this paper as SpaRSA (for **Spa**rse **R**econstruction by **S**eparable **A**pproximation). SpaRSA has the following desirable properties:

(a) when applied to the $\ell_2 - \ell_1$ problem (2), it is computationally competitive with the state-of-the-art algorithms designed specifically for that problem;
(b) it is versatile enough to handle a broad class of generalizations of (2), in which the ℓ_1 term is replaced with other regularization terms such as those described above;
(c) it is applicable immediately to problems (2) in which $\mathbf{A}$ and $\mathbf{y}$ (and hence the solution $\mathbf{x}$) contain complex data, as happens in many signal/image processing problems involving coherent observations, such as radar imaging or magnetic resonance imaging (MRI).

As mentioned above, our approach requires solution of (5) at each iteration. When the regularizer c is separable or group-separable, the solution of (5) can be obtained from a number of scalar (or otherwise low-dimensional) minimizations, whose solutions are often available in closed form. We discuss this issue further in Sections II-B and II-D.

The solution of (5) and (6) also solves the trust-region problem obtained by forming the obvious linear model of f around $\mathbf{x}^t$ and using an ℓ_2-norm constraint on the step, that

[2]The authors refer to this as an EM algorithm, which, in this case, is an IST algorithm; see [37].

is,

$$\begin{aligned} \min_{\mathbf{z}} \quad & \nabla f(\mathbf{x}^t)^T(\mathbf{z}-\mathbf{x}^t) + \tau c(\mathbf{z}) \\ \text{subject to} \quad & \|\mathbf{z}-\mathbf{x}^t\|_2 \leq \Delta_t, \end{aligned} \tag{10}$$

for some appropriate value of the trust-region radius Δ_t.

Different variants of the SpaRSA approach are distinguished by different choices of α_t. We are particularly interested in variants based on the formula proposed by Barzilai and Borwein (BB) [1] in the context of smooth nonlinear minimization; see also [19], [50]. Many variants of Barzilai and Borwein's approach, also known as *spectral methods*, have been proposed. They have also been applied to constrained problems [3], especially bound-constrained quadratic programs [18], [39], [68]. Pure spectral methods are nonmonotone; *i.e.*, the objective function is not guaranteed to decrease at every iteration; this fact makes convergence analysis a nontrivial task. We consider a so-called "safeguarded" version of SpaRSA, in which the objective is required to be slightly smaller than the largest objective in some recent past of iterations, and provide a proof of convergence for the resulting algorithm.

C. Related Work

Approaches related to SpaRSA have been investigated in numerous recent works. The recent paper of Figueiredo, Nowak, and Wright [39] describes the GPSR (*gradient projection for sparse reconstruction*) approach, which works with a bound-constrained reformulation of (2). Gradient-projection algorithms are applied to this formulation, including variants with spectral choices of the steplength parameters, and monotone and nonmonotone variants. When applied to $\ell_2-\ell_1$ problems, the SpaRSA approach of this paper is closely related to GPSR, but not identical to it. The steplength parameter in GPSR plays a similar role to the Hessian approximation term α_t in this paper. While matching the efficiency of GPSR on the $\ell_2-\ell_1$ case, SpaRSA can be generalized to a much wider class of problems, as described above.

SpaRSA is also closely related to *iterative shrinkage/thresholding* (IST) methods, which are also known in the literature by different names, such as *iterative denoising*, *thresholded Landweber*, *forward-backward splitting*, and *fixed-point iteration* algorithms (see Combettes and Wajs [17], Daubechies, Defriese, and De Mol [21], Elad [32], Figueiredo and Nowak [36], and Hale, Yin, and Zhang [51]). The form of the subproblem (5) is the same in these methods as in SpaRSA, but IST methods use a more conservative choice of α_t, related to the Lipschitz constant of ∇f. In fact, SpaRSA can be viewed as a kind of accelerated IST, with improved practical performance resulting from variation of α_t. Other ways to accelerate IST algorithms include two-step variants, as in the recently proposed *two-step IST* (TwIST) algorithm [2], continuation schemes (as suggested in the above mentioned [39] and [51], and explained in the next paragraph), and a semi-smooth Newton method [48]. Finally, we mention iterative coordinate descent (ICD) algorithms [8], [40], and block coordinate descent (BCD) algorithms [73]; those methods work by successively minimizing the objective with respect each component (or group of components) of $\mathbf{x}$, so are close in spirit to the well-known Gauss-Seidel (or block Gauss-Seidel) algorithms for linear systems.

The approaches discussed above, namely IST, SpaRSA, and GPSR, benefit from the use of a good approximate solution as a starting point. Hence, solutions to (2) and (1) can be obtained for a number of different values of the regularization parameter τ by using the solution calculated for one such value as a starting point for the algorithm to solve for a nearby value. It has been observed that the practical performance of GPSR, SpaRSA, IST, and other approaches degrades for small values of τ. Hale, Yin, and Zhang [51] recognized this fact and integrated a "continuation" procedure into their fixed-point iteration scheme, in which (2) is solved for a decreasing sequence of values of τ, using the computed solution for each value of τ as the starting point for the next smaller value. Using this approach, solutions are obtained for small τ values at much lower cost than if the algorithm was applied directly to (2) from a "cold" starting point. Similar continuation schemes have been implemented into GPSR [39, Section IV.D] and have largely overcome the computational difficulties associated with small regularization parameters. In this paper, we contribute further to the development of continuation strategies by proposing an adaptive scheme (suited to the $\ell_2-\ell_1$ case) which dispenses the user from having to define the sequence of values of τ to be used.

Van den Berg and Friedlander [75] have proposed a method for solving (4) for some $\varepsilon > 0$, by searching for the value of T for which the solution $\mathbf{x}^T$ of (3) has $\|\mathbf{y}-\mathbf{A}\mathbf{x}^T\|_2^2 = \varepsilon$. A rootfinding procedure is used to find the desired T, and the ability to solve (3) cheaply is needed. Yin *et al.* [77] have described a method for solving the basis pursuit problem, *i.e.*, (4) with $\varepsilon = 0$, where the main computational cost is the solution of a small number of problems of the form (2), for different values of $\mathbf{y}$ and possibly also τ. The technique is based on Bregman iterations and is equivalent to an augmented Lagrangian technique. SpaRSA can be used to efficiently solve each of the subproblems, since it is able to use the solution of one subproblem as a "warm start" for the next subproblem.

In a recent paper [61], Nesterov has presented three approaches, which solve the formulation (1) and make use of subproblems of the form (5). Nesterov's PG (primal gradient) approach follows the SpaRSA framework of Section II-A (and was in fact inspired by it), choosing the initial value of α_t at iteration t by modifying the final accepted value at iteration $t-1$, and using a "sufficient decrease" condition to test for acceptability of a step. Nesterov's other approaches, DG (a dual gradient method) and AC (an accelerated dual gradient approach), are less simple to describe. At each iteration, these methods solve a subproblem of the form (5) and a similar subproblem with a different linear and quadratic term; the next iteration is derived from both subproblems. Nesterov's computational tests on problems of the form (2) indicate that the most sophisticated variant, AC, is significantly faster than the other two variants.

Various other schemes have been proposed for the ℓ_2-ℓ_1 problem (2) and its alternative formulations (3) and (4. These include active-set-based homotopy algorithms [33], [56], [63]),

and interior-point methods [15], [67], [9], [10], [53]. Matching pursuit and orthogonal matching pursuit have also been proposed for finding sparse approximate solutions of $\mathbf{A}\mathbf{x} = \mathbf{y}$ [4], [23], [28], [72]; these methods, previously known in statistics as *forward selection* [76], are not based on an explicit optimization formulation. A more detailed discussion of those alternative approaches can be found in [39].

D. Outline of the Paper

Section II presents the SpaRSA framework formally, discussing how the subproblem in each iteration is solved (for several classes of regularizers) as well as the different alternatives for choosing parameter α_t; Section II also discusses stopping criteria and the so-called "debiasing" procedure. Section III presents an adaptive *continuation* scheme, which is empirically shown to considerably speed up the algorithm in problems where the regularization parameter is small. In Section IV, we report a series of experiments which show that SpaRSA has state of the art performance for the ℓ_2-ℓ_1 problems; other experiments described in that section illustrate that SpaRSA can handle a more general class of problems.

II. The Proposed Approach

A. The SpaRSA Framework

Rather than a specific algorithm, SpaRSA is an algorithmic framework for problems of the form (1), which can be instantiated by adopting different regularizers, different ways of choosing α_t, and different criteria to accept a solution to each subproblem (5). The SpaRSA framework is defined by the following pseudo-algorithm.

Algorithm *SpaRSA*

1. choose factor $\eta > 1$ and constants $\alpha_{\min}$, $\alpha_{\max}$ (with $0 < \alpha_{\min} < \alpha_{\max}$);
2. initialize iteration counter, $t \leftarrow 0$; choose initial guess $\mathbf{x}^0$;
3. **repeat**
4. choose $\alpha_t \in [\alpha_{\min}, \alpha_{\max}]$;
5. **repeat**
6. $\mathbf{x}^{t+1} \leftarrow$ solution of sub-problem (6);
7. $\alpha_t \leftarrow \eta\, \alpha_t$;
8. **until** $\mathbf{x}^{t+1}$ satisfies an acceptance criterion
9. $t \leftarrow t + 1$;
10. **until** stopping criterion is satisfied.

As mentioned above, the different instances of SpaRSA are obtained by making different design choices concerning two key steps of the algorithm: the setting of α_t (line 4) and the acceptance criterion (line 8). It is worth noting here that IST algorithms are instances of the SpaRSA framework. If c is convex (thus the subproblem (6) has a unique minimizer), if the acceptance criterion accepts any $\mathbf{x}^{t+1}$, and if we use a constant α_t satisfying certain conditions (see [17], for example), then we have a convergent IST algorithm.

B. Solving the Subproblems: Separable Regularizers

In this section, we consider the key operation of the SpaRSA framework — solution of the subproblem (6) — for situations in which the regularizer c is separable. Since the term $\|\mathbf{z}-\mathbf{u}^t\|_2^2$ is a strictly convex function of $\mathbf{z}$, (6) has a unique solution when c is convex. (For nonconvex c, there may exist several local minimizers.)

When c has the separable form (8), the subproblem (6) is also separable and can be written as

$$x_i^{t+1} \in \arg\min_z \ \frac{(z-u_i^t)^2}{2} + \frac{\tau}{\alpha_t}\, c_i(z), \quad i = 1, 2, \ldots, n. \tag{11}$$

For certain interesting choices of c_i, the minimization in (11) has a unique closed form solution. When $c(\mathbf{z}) = \|\mathbf{z}\|_1$ (thus $c_i(z) = |z|$), we have a unique minimizer given by

$$\arg\min_z \ \frac{(z-u_i^t)^2}{2} + \frac{\tau\,|z|}{\alpha_t} \ = \text{soft}\left(u_i^t, \frac{\tau}{\alpha_t}\right), \tag{12}$$

where $\text{soft}(u, a) \equiv \text{sign}(u)\max\{|u| - a, 0\}$ is the well-known soft-threshold function.

Another notable separable regularizer is the so-called ℓ_0 quasi-norm $c(\mathbf{z}) = \|\mathbf{z}\|_0 = \sum_i 1_{x_i \neq 0}$, which counts the number of nonzero components of its argument. Although $c_i(z) = 1_{z\neq 0}$ is not convex, there is a unique solution

$$\arg\min_z \ \frac{(z-u_i^t)^2}{2} + \frac{\tau}{\alpha_t}\, 1_{x_i\neq 0} = \text{hard}\left(u_i^t, \sqrt{\frac{2\,\tau}{\alpha_t}}\right), \tag{13}$$

where $\text{hard}(u, a) \equiv u\, 1_{|u|>a}$ is the hard-threshold function.

When $c_i(z) = |z|^p$, that is, $c(\mathbf{z}) = \|\mathbf{z}\|_p^p$, the closed form solution of (11) is known for $p \in \{4/3, 3/2, 2\}$ [14], [17]. For these values of p, the function c is convex and smooth. For $c_i(z) = |z|^p$ with $0 < p < 1$, the function c is nonconvex (though it is quasi-convex), but the solutions of (11) can still be obtained by applying a safeguarded Newton method and considering the cases $z < 0$, $z = 0$, and $z > 0$ separately.

C. Solving the Subproblems: the Complex Case

The extension of (2) to the case in which $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{y}$ are complex is more properly written as

$$\min_{\mathbf{x}\in\mathbb{C}^n} \ \frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{x})^H(\mathbf{y}-\mathbf{A}\mathbf{x}) + \tau\sum_{i=1}^n |x_i|, \tag{14}$$

where $|x_i|$ denotes the modulus of the complex number x_i. In this case, the subproblem (6) is

$$\mathbf{x}^{t+1} \in \arg\min_{\mathbf{z}\in\mathbb{C}^n} \ \frac{1}{2}(\mathbf{z}-\mathbf{u}^t)^H(\mathbf{z}-\mathbf{u}^t) + \frac{\tau}{\alpha_t}\sum_{i=1}^n |z_i|, \tag{15}$$

which is obviously still separable and leads to

$$\arg\min_{z\in\mathbb{C}} \ \frac{|z-u_i^t|^2}{2} + \frac{\tau\,|z|}{\alpha_t} \ = \text{soft}\left(u_i^t, \frac{\tau}{\alpha_t}\right), \tag{16}$$

with the (complex) soft-threshold function defined for complex argument by

$$\text{soft}(u, a) \equiv \frac{\max\{|u| - a, 0\}}{\max\{|u| - a, 0\} + a}\, u. \tag{17}$$

D. Solving the Subproblems: Group-Separable Regularizers

For group-separable (GS) regularizers of the form (9), the minimization (6) decouples into a set of m independent minimizations of the form

$$\min_{\mathbf{w}\in\mathbb{R}^l} \ \frac{1}{2}\,\|\mathbf{w}-\mathbf{b}\|_2^2 + \beta\,\Phi(\mathbf{w}), \tag{18}$$

where l is the dimension of $\mathbf{x}_{[i]}$, $\mathbf{b} = \mathbf{u}^t_{[i]}$, $\Phi = c_i$, and $\beta = \tau/\alpha_t$, with $\mathbf{u}^t$ defined in (7).

As in [14], [17], convex analysis can be used to obtain the solution of (18). If Φ is a norm, it is proper, convex (though not necessarily strictly convex), and homogenous. Since, in addition, the quadratic term in (18) is proper and strictly convex, this problem has a unique solution, which can be written explicitly as

$$\arg\min_{\mathbf{w}\in\mathbb{R}^l} \ \frac{1}{2}\,\|\mathbf{w}-\mathbf{b}\|_2^2 + \beta\,\Phi(\mathbf{w}) = \mathbf{b} - P_{\beta C_\Phi}(\mathbf{b}), \tag{19}$$

where P_B denotes the orthogonal projector onto set B, and C_Φ is a unit-radius ball in the dual norm $\Phi^\star$, that is, $C_\Phi = \{\mathbf{w}\in\mathbb{R}^l : \Phi^\star(\mathbf{w}) \le 1\}$. Detailed proofs of (19) can be found in [17] and references therein.

Taking Φ as the ℓ_2 or ℓ_∞ norm is of particular interest in the applications mentioned above. For $\Phi(\mathbf{w}) = \|\mathbf{w}\|_2$, the dual norm is also $\Phi^\star(\mathbf{w}) = \|\mathbf{w}\|_2$, thus $\beta C_{\|\cdot\|_2} = \{\mathbf{w}\in\mathbb{R}^l : \|\mathbf{w}\|_2 \le \beta\}$. Clearly, if $\|\mathbf{b}\|_2 \le \beta$, then $P_{\beta C_{\|\cdot\|_2}}(\mathbf{b}) = \mathbf{b}$, thus $\mathbf{b} - P_{\beta C_{\|\cdot\|_2}}(\mathbf{b}) = 0$. If $\|\mathbf{b}\|_2 > \beta$, then $P_{\beta C_{\|\cdot\|_2}}(\mathbf{b}) = \beta\,\mathbf{b}/\|\mathbf{b}\|_2$. These two cases are written compactly as

$$\mathbf{w} = \mathbf{b}\,\frac{\max\{\|\mathbf{b}\|_2-\beta,0\}}{\max\{\|\mathbf{b}\|_2-\beta,0\}+\beta}, \tag{20}$$

which can be seen as a vectorial soft-threshold. Naturally, if $l = 1$, (20) reduces to the scalar soft-threshold (12).

For $\Phi(\mathbf{w}) = \|\mathbf{w}\|_\infty$, the dual norm is $\Phi^\star(\mathbf{w}) = \|\mathbf{w}\|_1$, thus $\beta C_{\|\cdot\|_\infty} = \{\mathbf{w}\in\mathbb{R}^n : \|\mathbf{w}\|_1 \le \beta\}$. In this case, the solution of (18) is the residual of the orthogonal projection of $\mathbf{b}$ onto the ℓ_1 β-ball. This projection can be computed with $O(l\log l)$ cost, as recently shown in [5], [6], [7], [22]; even more recently, an $O(l)$ algorithm was introduced [31].

E. Choosing α_t: Barzilai-Borwein (Spectral) Methods.

In the most basic variant of the Barzilai-Borwein (BB) spectral approach, α_t is chosen such that $\alpha_t\,\mathbf{I}$ mimics the Hessian $\nabla^2 f(\mathbf{x})$ over the most recent step. Letting $\mathbf{s}^t = \mathbf{x}^t - \mathbf{x}^{t-1}$ and

$$\mathbf{r}^t = \nabla f(\mathbf{x}^t) - \nabla f(\mathbf{x}^{t-1}),$$

we require that $\alpha_t\,\mathbf{s}^t \approx \mathbf{r}^t$ in the least-squares sense, *i.e.*,

$$\alpha_t = \arg\min_\alpha \|\alpha\,\mathbf{s}^t - \mathbf{r}^t\|_2^2 = \frac{(\mathbf{s}^t)^T\mathbf{r}^t}{(\mathbf{s}^t)^T\mathbf{s}^t}. \tag{21}$$

When $f(\mathbf{x}) = (1/2)\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2$, this expression becomes $\alpha_t = \|\mathbf{A}\,\mathbf{s}^t\|_2^2/\|\mathbf{s}^t\|_2^2$. In our implementation of the SpaRSA framework, we use (21) to choose the first α_t in each iteration (line 4 of Algorithm *SpaRSA*), safeguarded to ensure that α_t remains in the range $[\alpha_{\min}, \alpha_{\max}]$.

A similar approach, also suggested by Barzilai and Borwein [1] is to choose β_t so that $\beta_t\mathbf{I}$ mimics the behavior of the *inverse* Hessian over the latest step, and then set $\alpha_t = \beta_t^{-1}$. By solving $\mathbf{s}^t = \beta_t\mathbf{r}^t$ in the least-squares sense, we obtain

$$\alpha_t = \frac{(\mathbf{r}^t)^T\mathbf{r}^t}{(\mathbf{r}^t)^T\mathbf{s}^t} = \frac{\|\mathbf{A}^T\mathbf{A}\mathbf{s}^t\|^2}{\|\mathbf{A}\mathbf{s}^t\|^2}.$$

Other spectral methods have been proposed that alternate between these two formulae for α_t. There are also "cyclic" variants in which α_t is only updated (using the formulae above) at every S-th iteration ($S\in\mathbb{N}$); see Dai *et al.* [19]. We will not consider those variants in this paper, since we have verified experimentally that their performance is very close to that of the standard BB method based on (21).

F. Acceptance Criterion

In the simplest variant of SpaRSA, the criterion used at each iteration to decide whether to accept a candidate step is trivial: accept whatever $\mathbf{z}$ solves the subproblem (5) as the new iterate $\mathbf{x}^{t+1}$, even if it yields an increase in the objective function ϕ. Barzilai-Borwein schemes are usually implemented in this nonmonotone fashion. The drawback of these totally "free" BB schemes is that convergence is very hard to study.

Globally convergent Barzilai-Borwein schemes for unconstrained smooth minimization have been proposed in which the objective is required to be slightly smaller than the largest objective from the last M iterations, where M is a fixed integer (see [50]). If M is chosen large enough, the occasional large increases in objective (that are characteristic of BB schemes, and that appear to be essential to their good performance in many instances) are still allowed. Inspired by this observation, we propose an acceptance criterion in which the candidate $\mathbf{x}^{t+1}$ obtained in line 6 of the algorithm (a solution of (6)) is accepted as the new iterate if its objective value is slightly smaller than the largest value of the objective ϕ over the past $M+1$ iterations. Specifically, $\mathbf{x}^{t+1}$ is accepted only if

$$\phi(\mathbf{x}^{t+1}) \le \max_{i=\max(t-M,0),\ldots,t} \phi(\mathbf{x}^i) - \frac{\sigma}{2}\,\alpha_t\,\|\mathbf{x}^{t+1}-\mathbf{x}^t\|^2, \tag{22}$$

where $\sigma\in(0,1)$ is a constant, usually chosen to be close to zero. This is the version of the proposed algorithmic framework which we will simply denote as SpaRSA.

We consider also a monotone version (called SpaRSA-monotone) which is obtained by letting $M = 0$. The existence of a value of α_t sufficiently large to ensure a decrease in the objective at each iteration can be inferred from the connection between (6) and the trust-region subproblem (10). For a small enough trust-region radius Δ_t, the difference between the linearized model in (10) and the true function $\phi(\mathbf{z}) - \phi(\mathbf{x}^t)$ becomes insignificant, so the solution of (10) is sure to produce a decrease in ϕ. Monotonicity of IST algorithms [37] also relies on the fact that there is a constant $\bar\alpha > 0$ such that descent is assured whenever $\alpha_t \ge \bar\alpha$.

G. Convergence

We now present a global convergence result for SpaRSA applied to problems with the form of (1), with a few mild conditions, which are satisfied by essentially all problems of

interest. Specifically, we assume that f is Lipschitz continuously differentiable, that c is convex and finite valued, and that ϕ is bounded below.

Before stating the theorem, we recall that a point $\bar{\mathbf{x}}$ is said to be *critical* for (1) if

$$0 \in \partial\phi(\bar{\mathbf{x}}) = \nabla f(\bar{\mathbf{x}}) + \tau \partial c(\bar{\mathbf{x}}). \tag{23}$$

where ∂c denotes the subdifferential of c (see [65] for a definition). Criticality is a necessary condition for optimality. When f is convex, then ϕ is convex also, and condition (23) is sufficient for $\bar{\mathbf{x}}$ to be a global solution of (1). Our theorem shows that all accumulation points of SpaRSA are critical points, and therefore global solutions of (1) when f is convex.

Theorem 1: Suppose that Algorithm SpaRSA, with acceptance test (22), is applied to (1), where f is Lipschitz continuously differentiable, c is convex and finite-valued, and ϕ is bounded below. Then all accumulation points are critical points.

The proof, which can be found in the Appendix, is inspired by the work of Grippo, Lampariello, and Lucidi [49], who analyzed a nonmonotone line-search Newton method for optimization of a smooth function whose acceptance condition is analogous to (22).

H. Termination Criteria

We described a number of termination criteria for GPSR in [39]. Most of these continue to apply in SpaRSA, in the case of $c(\mathbf{x}) = \|\mathbf{x}\|_1$. We describe them briefly here, and refer the reader to [39, Subsection II-D] for further details.

One termination criterion for (2) can be obtained by reformulating it as a linear complementarity problem (LCP). This is done by splitting $\mathbf{x}$ as $\mathbf{x} = \mathbf{v} - \mathbf{w}$ with $\mathbf{v} \geq 0$ and $\mathbf{w} \geq 0$, and writing the equivalent problem as

$$\min \left(\begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix}, \begin{bmatrix} \tau \mathbf{1}_n + \mathbf{A}^T(\mathbf{A}(\mathbf{v}-\mathbf{w}) - \mathbf{y}) \\ \tau \mathbf{1}_n - \mathbf{A}^T(\mathbf{A}(\mathbf{v}-\mathbf{w}) - \mathbf{y}) \end{bmatrix} \right) = 0, \tag{24}$$

where $\mathbf{1}_n$ is the vector of 1s with length n, and the minimum is taken componentwise. The distance to the LCP solution set from a given vector $(\mathbf{v}, \mathbf{w})$ is bounded by a multiple of the norm of the left-hand side in (24), so it is reasonable to terminate when this quantity falls below a given small tolerance `tolP`, where we set $\mathbf{v} = \max(\mathbf{x}, 0)$ and $\mathbf{w} = \max(-\mathbf{x}, 0)$.

Another criterion for the problem (2) can be obtained by finding a feasible point $\mathbf{s}$ for the dual of this problem, which can be written as

$$\max_{\mathbf{s}} -\frac{1}{2}\mathbf{s}^T\mathbf{s} - \mathbf{y}^T\mathbf{s}, \text{ subject to } -\tau \mathbf{1}_n \leq \mathbf{A}^T\mathbf{s} \leq \tau \mathbf{1}_n,$$

and then finding the duality gap corresponding to $\mathbf{s}$ and the current primal iterate $\mathbf{x}^t$. This quantity yields an upper bound on the difference between $\phi(\mathbf{x}^t)$ and the optimal objective value ϕ^*, so we terminate when the relative duality gap falls below a tolerance `tolP`. Further details can be found in [39, Subsection II-D] and [53].

We note too that the criterion based on the relative change to the set of inactive indices $\mathcal{I}_t := \{i = 1, 2, \ldots, n \,|\, \mathbf{x}_i^t \neq 0\}$ between iterations, can also be applied. The technique described in [39, Subsection II-D] can be extended by monitoring the change in inactive set across a range of steps, not just the single previous step from $\mathbf{x}^{t-1}$ to $\mathbf{x}^t$. It can also be extended to group-separable problems by defining the inactive set in terms of groups rather than individual components.

A less sophisticated criterion makes use of the relative change in objective value at the last step. We terminate at iteration t if

$$\frac{|\phi(\mathbf{x}^t) - \phi(\mathbf{x}^{t-1})|}{\phi(\mathbf{x}^{t-1})} \leq \texttt{tolP}. \tag{25}$$

This criterion has the advantage of generality; it can be used for any choice of regularization function c. However, it is problematic to use in general as it may be triggered when the step between the last two iterates was poor, but the current point is still far from a solution. When used in the context of nonmonotone methods it is particularly questionable, as steps that produce a significant decrease *or increase* in ϕ are deemed acceptable, while those which produce little change in ϕ trigger termination. Still, we have rarely ecountered problems of "false termination" with this criterion in our computational tests.

A similarly simple and general criterion is the relative size of the step just taken, that is,

$$\frac{\|\mathbf{x}^t - \mathbf{x}^{t-1}\|}{\|\mathbf{x}^t\|} \leq \texttt{tolP}. \tag{26}$$

This criterion has some of the same possible pitfalls as (25), but again we have rarely observed it to produce false termination provided `tolP` is chosen sufficiently small.

When a continuation strategy (Subsection III) is used, in which we we do not need the solutions for intermediate values of τ to high accuracy, we can use a tight criterion for the final value of τ and different (and looser) criteria for the intermediate values. In our implementation of SpaRSA, we used the criterion (25) with $\texttt{tolP} = 10^{-5}$ at the intermediate stages, and switched to the criterion specified by the user for the target value of τ.

Finally, we make the general comment that termination at solutions that are "accurate enough" for the application at hand while not being highly accurate solutions of the optimization problem is an issue that has been litte studied by optimization specialists. It is usually (and perhaps inevitably) left to the user to tune the stopping criteria in their codes to the needs of their application. This issue is perhaps deserving of study at a more general level, as the choice of stopping criteria can dramatically affect the performance of many optimization algorithms in practice.

I. Debiasing

In many situations, it is worthwhile to debias the solution as a postprocessing step, to eliminate the attenuation of signal magnitude due to the presence of the regularization term. In the debiasing step, we fix at zero those individual components (in the case of ℓ_1 regularization) or groups (in the case of group regularization) that are zero at the end of the SpaRSA process, and minimize the objective f over the remaining

elements. Specifically, the case of a sum-of-squares objective $(1/2)\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2$, the debiasing phase solves the problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}_{\cdot\mathcal{I}}\mathbf{x}_{\mathcal{I}}-\mathbf{y}\|_2^2, \tag{27}$$

where $\mathcal{I}$ is the set of indices corresponding to the components or groups that were nonzero at termination of the SpaRSA procedure for minimizing ϕ, $\mathbf{A}_{\cdot\mathcal{I}}$ is the column submatrix of $\mathbf{A}$ corresponding to $\mathcal{I}$, and $\mathbf{x}_{\mathcal{I}}$ is the subvector of unknowns for this index set. A conjugate gradient procedure is used, and the debiasing phase is terminated when the squared residual norm for (27), that is

$$\left\|\mathbf{A}_{\cdot\mathcal{I}}^T(\mathbf{A}_{\cdot\mathcal{I}}\mathbf{x}_{\mathcal{I}}-\mathbf{y})\right\|_2^2,$$

falls below its value at the SpaRSA solution by a factor of `tolD`, where a typical value is `tolD` $= 10^{-4}$. (The same criterion is used in GPSR; the criterion shown in [39, (21)] is erroneous.) When the column submatrix $\mathbf{A}_{\cdot\mathcal{I}}$ is well conditioned, as happens when a restricted isometry property is satisfied, the conjugate gradient procedure converges quite rapidly, consistently with the known theory for this method (see for example Golub and Van Loan [46, Section 10.2]).

It was shown in [39], for example, that debiasing can improve the quality of the recovered signal considerably. Such is not always the case, however. Shrinking of signal coefficients can sometimes have the desirable effect of reducing distortions caused by noise [26], an effect that could be undone by debiasing.

III. Warm Starting and Adaptive Continuation

Just as for the GPSR and IST algorithms, the SpaRSA approach benefits significantly from a good starting point $\mathbf{x}^0$, which suggests that we can use the solution of (1), for a given value of τ, to initialize SpaRSA in solving (1) for a nearby value of τ. Generally, the "warm-started" second run will require fewer iterations than the first run, and dramatically fewer iterations than if it were initialized at zero.

An important application of warm-starting is *continuation*, as in the *fixed point continuation* (FPC) algorithm recently described in [51]. It has been observed that IST, SpaRSA, GPSR, and other approaches become slow when applied to problems (2) with small values of the regularization parameter τ. (Solving (2) with a very small value of τ is one way of approximately solving (4) with $\varepsilon = 0$.) However, if we use SpaRSA to solve (1) for a larger value of τ, then decrease τ in steps toward its desired value, running SpaRSA with warm-start for each successive value of τ, we are often able to identify the solution much more efficiently than if we just ran SpaRSA once for the desired (small) value of τ from a cold start. We illustrate this claim experimentally in Section IV.

One of the challenges in using continuation is to choose the sequence of τ values that leads to the fastest global running time. In the continuation schemes proposed in [51] and [39], it is left to the user to define this sequence of τ values. Here we propose a scheme for the $\ell_2-\ell_1$ case that does not require the user to specify the sequence of values of τ. Our adaptive scheme is based on the fact that it is possible to give some meaning to the notions of "large" or "small", when referring to the regularization parameter τ in the context of problem (2). It can be shown that if

$$\tau \geq \|\mathbf{A}^T\mathbf{y}\|_\infty,$$

then the unique solution to (2) is the zero vector [41], [53]. Accordingly, a value of τ such that $\tau \lesssim \|\mathbf{A}^T\mathbf{y}\|_\infty$ can be considered "large", while a value such that $\tau \ll \|\mathbf{A}^T\mathbf{y}\|_\infty$ can be seen as small. Inspired by this fact, we propose the following scheme for solving (2):

Algorithm *Adaptive Continuation*

1. initialize iteration counter, $t \leftarrow 0$, and choose initial estimate $\mathbf{x}^0$;
2. $\mathbf{y}^t \leftarrow \mathbf{y}$;
3. **repeat**
4. $\quad\tau_t \leftarrow \max\{\zeta\|\mathbf{A}^T\mathbf{y}^t\|_\infty, \tau\}$, where $\zeta < 1$;
5. $\quad\mathbf{x}^{t+1} \leftarrow \text{SpaRSA}(\mathbf{y}, \mathbf{A}, \tau_t, \mathbf{x}^t)$;
6. $\quad\mathbf{y}^{t+1} \leftarrow \mathbf{y} - \mathbf{A}\mathbf{x}^{t+1}$;
7. $\quad t \leftarrow t+1$;
8. **until** $\tau_t = \tau$;

In line 5 of the algorithm, $\text{SpaRSA}(\mathbf{y}, \mathbf{A}, \tau_t, \mathbf{x}^t)$ denotes a run of the SpaRSA algorithm for problem (2), with τ replaced by τ_t, and initialized at $\mathbf{x}^t$. The key steps of the algorithm are those in lines 4, 5, and 6, and the rationale behind these steps is as follows. After running SpaRSA with the regularization parameter τ_t, the linear combination of the columns of $\mathbf{A}$, according to the latest iterate $\mathbf{x}^{t+1}$, is subtracted from the observation $\mathbf{y}$, yielding $\mathbf{y}^{t+1}$. The idea is that $\mathbf{y}^{t+1}$ contains the information about the unknown $\mathbf{x}$ which can only be obtained with a smaller value of the regularization parameter; moreover, the "right" value of the regularization parameter to extract some more of this information is given by the expression in line 4 of the algorithm. Notice that in step 5, SpaRSA is always run with the original observed vector $\mathbf{y}$ (not with $\mathbf{y}^t$), so our scheme is not a pursuit-type method (such as StOMP [30]).

We note that if the invocation of SpaRSA in line 5 produces an exact solution, we have that $\|\mathbf{A}^T\mathbf{y}^{t+1}\|_\infty = \tau_t$, so that line 4 simply reduces the value of τ by a constant factor of ζ at each iteration. Since in practice an exact solution may not be obtained in line 5, the scheme above produces different computational behavior which is usually better in practice. Although the description of the adaptive continuation scheme was made with reference to the SpaRSA algorithm, this scheme can be used with any other algorithm that benefits from good initialization and that is faster for larger values of the regularization parameter. For example, by using IST in place of SpaRSA in line 5, we obtain an adaptive version of the FPC algorithm [51].

IV. Computational Experiments

In this section we report experiments which demonstrate the competitive performance of the SpaRSA approach on problems of the form (2), including problems with complex data, and its ability to handle different types of regularizers. All the experiments (except for those in Subsection IV-E) were carried out on a personal computer with an Intel *Core2Extreme* 3 GHz processor and 4GB of memory, using a MATLAB

implementation of SpaRSA. The parameters of SpaRSA were set as follows: $M = 5$, $\sigma = 0.01$, $\alpha_{\max} = 1/\alpha_{\min} = 10^{30}$; for SpaRSA-monotone, we set $M = 0$, $\sigma = 10^{-5}$, and $\eta = 2$; finally, for the adaptive continuation strategy, we set $\zeta = 0.2$.

A. Speed Comparisons for the $\ell_2 - \ell_1$ Problem

We compare the performance of SpaRSA with that of other recently proposed algorithms for $\ell_2 - \ell_1$ problems (2). In our first experiment, in addition to the monotone and nonmonotone variants of SpaRSA, we consider the following algorithms: GPSR [39], FPC [51], TwIST [2], `l1_ls` [53], and AC [61]. The $\ell_2 - \ell_1$ test problem that we consider is similar to the one studied in [53] and [39]. The matrix $\mathbf{A}$ in (2) is a random $k \times n$ matrix, with $k = 2^{10}$ and $n = 2^{12}$, with Gaussian i.i.d. entries of zero mean and variance $1/(2n)$. (This variance guarantees that, with high probability, the maximum singular value of $\mathbf{A}$ is at most 1, which is assumed by FPC and TwIST.) We choose $\mathbf{y} = \mathbf{A}\mathbf{x}_{\text{true}} + \mathbf{e}$, where $\mathbf{e}$ is a Gaussian white vector with variance 10^{-4}, and $\mathbf{x}_{\text{true}}$ is a vector with 160 randomly placed ± 1 spikes, with zeros in the other components. We set $\tau = 0.1 \, \|\mathbf{A}^T\mathbf{y}\|_\infty$, as in [39], [53]; this value allows the $\ell_2 - \ell_1$ formulation to recover the solution, to high accuracy.

To make the comparison independent of the stopping rule for each approach, we first run FPC to set a benchmark objective value, then run the other algorithms until they each reach this benchmark. Table I reports the CPU times required by the algorithms tested, as well as the final mean squared error (MSE) of the reconstructions with respect to $\mathbf{x}_{\text{true}}$. These results show that, for this $\ell_2 - \ell_1$ problem, SpaRSA is slightly faster than GPSR and TwIST, and clearly faster than FPC, `l1_ls`, and AC. Not surprisingly, given that all approaches attain a similar final value of ϕ, they all give a similar value of MSE. Of course, these speed comparisons are implementation dependent, and should not be considered as a rigorous test, but rather as an indication of the relative performance of the algorithms for this class of problems.

One additional one order of magnitude improvement in MSE can be obtained easily by using the debiasing procedure described in Subsection II-I. In this problems, this debiasing step takes (approximately) an extra 0.15 seconds.

An indirect comparison with other codes can be made via [53, Table 1], which shows that `l1_ls` outperforms the method from [29] by a factor of approximately two, as well as ℓ_1-magic by about two orders of magnitude and `pdco` from `SparseLab` by about one order of magnitude.

The second experiment assesses how the computational cost of SpaRSA grows with the size of matrix $\mathbf{A}$, using a setup similar to the one in [39], [53]. We assume that the computational cost is $O(n^\alpha)$ and obtain empirical estimates of the exponent α. We consider random sparse matrices (with the nonzero entries normally distributed) of dimensions $(0.1\,n) \times n$, with n ranging from 10^4 to 10^6. Each matrix is generated with about $3n$ nonzero elements and the original signal with $n/4$ randomly placed nonzero components. For each value of n, we generate 10 random matrices $\mathbf{A}$ and original signals $\mathbf{x}$ and observed data according to $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is white noise of variance $\sigma^2 = 10^{-4}$. For each data set (that is, each pair $\mathbf{A}$, $\mathbf{y}$), τ is set to $0.1 \, \|\mathbf{A}^T\mathbf{y}\|_\infty$. The results in Fig. 1 (which are averaged over the 10 data sets of each size) show that SpaRSA, GPSR, and FPC have approximately linear cost, with FPC being a little worse than the other two algorithms. The exponent for `l1_ls` is known from [39], [53] to be approximately 1.2, while that of the ℓ_1-magic algorithms is approximately 1.3.

TABLE I
CPU TIMES AND MSE VALUES (AVERAGE OVER 10 RUNS) OF SEVERAL ALGORITHMS ON THE EXPERIMENT DESCRIBED IN THE TEXT; THE FINAL VALUE OF THE OBJECTIVE FUNCTION IS THE APPROXIMATELY 3.635 FOR ALL METHODS.

Algorithm	CPU time (secs.)	MSE
SpaRSA	0.32	3.42e-3
SpaRSA-monotone	0.34	3.43e-3
GPSR-BB-monotone	0.43	3.45e-3
GPSR-Basic	0.63	3.42e-3
FPC	1.52	3.44e-3
`l1_ls`	6.57	3.43e-3
AC	2.89	3.46e-3
TwIST	0.57	3.43e-3

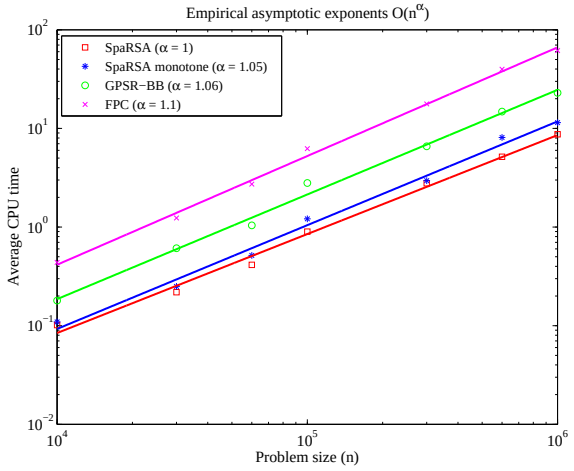

Fig. 1. Assessment of the empirical growth exponent of the computational complexity of several algorithms.

B. Adaptive Continuation

To assess the effectiveness of the adaptive regularization scheme proposed in Section III, we consider a scenario similar to the one in the first experiment, but with two differences. The data is noiseless, that is, $\mathbf{y} = \mathbf{A}\mathbf{x}_{\text{true}}$, and the regularization parameter is set to $\tau = 0.001 \, \|\mathbf{A}^T\mathbf{y}\|_\infty$. The results shown in Table II confirm that, with this small value of the regularization parameter, both GPSR and SpaRSA without continuation become significantly slower and that continuation yields a significant speed improvement. (We do not implement the continuation strategy for `l1_ls` as, being an interior point method, it does not benefit greatly from warm starts.) In this example, the debiasing step of Section II-I takes about 0.15 seconds, and yields an additional reduction in MSE by a factor of approximately 15.

The plot in Figure 2 shows how the CPU time of SpaRSA with and without continuation (as well as GSPR and FPC)

TABLE II
CPU TIMES AND MSE VALUES (AVERAGE OVER 10 RUNS) OF SEVERAL ALGORITHMS, WITHOUT AND WITH CONTINUATION, ON THE EXPERIMENT DESCRIBED IN THE TEXT. NOTICE THAT FPC HAS BUILT-IN CONTINUATION, SO IT IS LISTED IN THE CONTINUATION METHODS COLUMN.

Algorithm	CPU time (secs.), no continuation	CPU time (secs.), continuation	MSE
SpaRSA	16.18	1.61	4.96e-7
SpaRSA-monot.	17.13	1.63	3.41e-7
GPSR-BB-monot.	26.38	2.01	5.39e-7
GPSR-Basic	43.17	1.88	4.86e-7
FPC	–	5.22	5.49e-7
`l1_ls`	28.24	–	8.51e-7
AC	32.30	10.84	5.31e-7
TwIST	3.53	–	4.59e-7

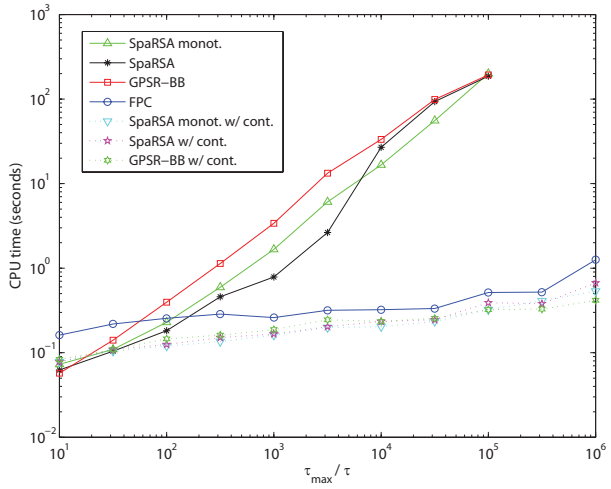

Fig. 2. CPU times as a function of the ratio $\tau/\tau_{\max}$, where $\tau_{\max} = \|\mathbf{A}^T\|_\infty$, for several algorithms without and with continuation.

grows when the value of the regularization parameter decreases, confirming that continuation is able to keep this growth very mild, in contrast to the behavior without continuation.

C. Group-Separable Regularizers

We now illustrate the use of SpaRSA with the GS regularizers defined in (9). Our experiments in this subsection use synthetic data and are mainly designed to illustrate the difference between reconstructions obtained with the GS-ℓ_2 and the GS-ℓ_∞ regularizers, both of which can be solved in the SpaRSA framework. In Subsection IV-E below, we describe experiments with GS regularizers, using magnetoencephalographic (MEG) data.

Our first synthetic experiment uses a matrix $\mathbf{A}$ with the same dimension and structure as the matrix in Subsection IV-A. The vector $\mathbf{x}_{\text{true}}$ has 2^{12} components, divided into $m = 64$ groups of length $l_i = 64$. To generate $\mathbf{x}_{\text{true}}$, we randomly choose 8 groups and fill them with zero-mean Gaussian random samples of unit variance, while all other groups are filled with zeros. We set $\mathbf{y} = \mathbf{A}\mathbf{x}_{\text{true}} + \mathbf{e}$, where $\mathbf{e}$ is Gaussian white noise with variance 10^{-4}. Finally we run SpaRSA, with $f(\mathbf{x}) = (1/2)\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$ and $c(\mathbf{x})$ as given by (9), where $c_i(\mathbf{x}_{[i]}) = \|\mathbf{x}_{[i]}\|_2$. The value of τ is hand-tuned for optimal performance. Figure 3 shows the result obtained by SpaRSA, based on the GS-ℓ_2 regularizer, which successfully recovers the group structure of $\mathbf{x}_{\text{true}}$, as well as the result obtained with the classical ℓ_1 regularizer, for the best choice of τ. The improvement in reconstruction quality obtained by exploiting the known group structure is evident.

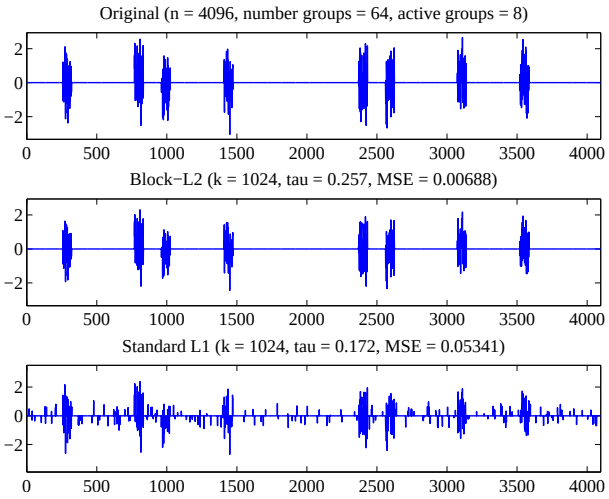

Fig. 3. Comparison of GS-ℓ_2 regularizer with a conventional ℓ_1 regularizer. This example illustrates how exploiting known group structure can provide a dramatic gain.

In the second experiment, we consider a similar scenario, with a single difference: Each active group, instead of being filled with Gaussian random samples, is filled with ones. This case is clearly more adequate for a GS-ℓ_∞ regularizer, as illustrated in Figure 4, which achieves an almost perfect reconstruction, with an MSE two orders of magnitude smaller than the MSE obtained with a GS-ℓ_2 regularizer.

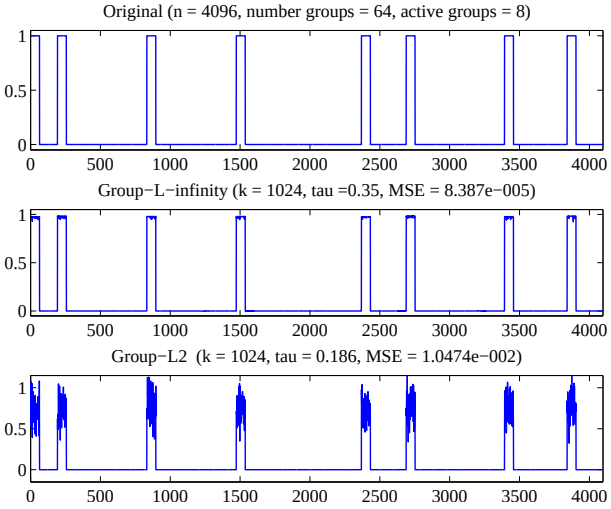

Fig. 4. Comparison of GS-ℓ_2 and GS-ℓ_∞ regularizers. Signals with uniform behavior within groups benefit from the GS-ℓ_∞ regularizer.

D. Problems with Complex Data

SpaRSA — like IST, FPC, ICD, and TwIST, but unlike GPSR — can be applied to complex data, provided that the regularizer is such that the subproblem at each iteration allows a simple solution. We illustrate this possibility by considering a classical signal processing problem where the goal is to estimate the number, amplitude, and initial phase of a set of superimposed sinusoids, observed under noise [25], [42], a problem that arises, for example, in *direction of arrival* (DOA) estimation [47] and spectral analysis [8]. Several authors have addressed this problem as that of estimating of sparse complex vector [8], [42], [47].

A discrete formulation of this problem may be given the form (2), where matrix $\mathbf{A}$ is complex, of size $k \times 2m_f$ (where m_f is the maximum frequency), with elements given by

$$A_{jf} = \exp\{\mathrm{i}\, 2\,\pi\, j\, f/(2m_f)\}, \quad \text{for } f = 1, ..., m_f, \quad j = 1, ..., k,$$

and $A_{jf} = A^*_{i(j-m_f)}$, for $f = m_f + 1, ..., 2m_f$ and $j = 1, ..., k$. As usual, i denotes $\sqrt{-1}$. For further details, see [42]. It is assumed that the observed signal is generated according to

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{28}$$

where $\mathbf{x}$ is a $2m_f$-vector in which $x_{f+m_f} = x^*_f$, for $f = 1, ..., m_f$, with four random complex entries appearing in four random locations among the first m_f elements. Each sinusoid is represented by two (conjugate) components of $\mathbf{x}$, that is, $x_f = A_f e^{\phi_f \mathbf{i}}$ and $x_{f+m_f} = x^*_f = A_f e^{-\phi_i \mathbf{i}}$, where A_f is its amplitude and ϕ_f its initial phase. The noise vector $\mathbf{n}$ is a vector of i.i.d. samples of complex Gaussian noise with standard deviation .05.

The noisy signal, the clean original signal (obtained by (28), without noise) and its estimate are shown in Figure 5. These results show that the $\ell_2 - \ell_1$ formulation and the SpaRSA and FPC algorithms are able to handle this problem. In this example, SpaRSA (with adaptive continuation) converges in 0.56 seconds, while the FPC algorithm obtains a similar result in 1.43 seconds.

E. MEG Brain Imaging

To see how our approach can speed up real-world optimization problems, we applied variants of SpaRSA to a magnetoencephalographic (MEG) brain imaging problem, replacing the EM algorithm of [5], [6], [7], which is equivalent to IST. MEG imaging using sparseness-inducing regularization was also previously considered in [47].

In MEG imaging, very weak magnetic fields produced by neuronal activity in the cortex are measured and used to infer cortical activity. The physics of the problem lead to an underdetermined linear model relating cortical activity from tens of thousands of voxels to measured magnetic fields at 100 to 200 sensors. This model combined with low SNR necessitates regularization of the inverse problem.

We solve the GS-ℓ_2 version of the regularization problem where each block of coefficients corresponds to a spatio-temporal subspace. The spatial components of each block describe the measurable activity within a local region of the cortex, while the temporal components describe low frequency activity in various time windows. The cortical activity inference problem is formulated as

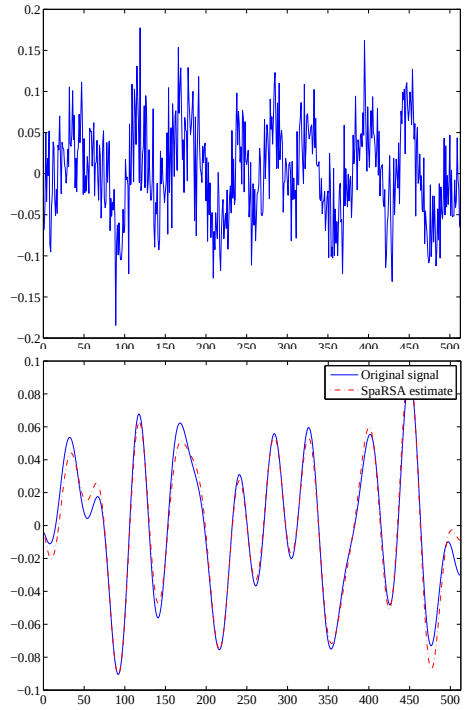

Fig. 5. Top plot: noisy superposition of four sinusoidal functions. Bottom plot: the original (noise free) superposition and its SpaRSA estimate.

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \|\mathbf{Y} - \mathbf{A}\mathbf{S}\mathbf{\Theta}\mathbf{T}^T\|_F^2 + \lambda \sum_{i,j} \|\mathbf{\Theta}_{i,j}\|_F, \tag{29}$$

where $\mathbb{Y}$ is the $k \times m$ matrix of the length-m time signals recorded at each of the k sensors, where $\mathbf{A}$ is the $k \times n$ linear mapping of cortical activity to the sensors, $\mathbf{S}$ is the dictionary of spatial bases, $\mathbf{T}$ is the dictionary of temporal bases, and $\mathbf{\Theta}$ contains the unknown coefficients that represent the cortical activity in terms of the spatio-temporal basis. Both $\mathbf{S}$ and $\mathbf{T}$ are organized into blocks of coefficients likely to be active simultaneously. The blocks of coefficients $\mathbf{\Theta}_{i,j}$ represent individual *space-time events* (STEs). The estimate of cortical activity is the sum of a small number of active (nonzero) STEs,

$$\hat{\mathbf{X}} = \sum_{i,j} \mathbf{S}_i \mathbf{\Theta}_{i,j} \mathbf{T}_j^T, \tag{30}$$

where most $\mathbf{\Theta}_{i,j}$ are zero.

An EM algorithm to solve the optimization above was proposed in [5], [6], [7]; that EM algorithm works by repeating two basic steps,

$$\begin{aligned} \hat{\mathbf{Z}}^{(t)} &= \hat{\mathbf{\Theta}}^{(t-1)} + c\mathbf{S}^T\mathbf{H}^T(\mathbf{Y} - \mathbf{A}\mathbf{S}\hat{\mathbf{\Theta}}^{(t-1)}\mathbf{T}^T)\mathbf{T} \\ \hat{\mathbf{\Theta}}^{(t)} &= \arg\min_{\mathbf{\Theta}} \left\{ \|\mathbf{\Theta} - \hat{\mathbf{Z}}^{(t)}\|_F^2 + c\lambda \sum_{i,j} \|\mathbf{\Theta}_{i,j}\|_F \right\}, \end{aligned} \tag{31}$$

where c is a step size. It is not difficult to see that this approach fits the SpaRSA framework, with subproblems of the

form (18), for a constant choice of parameter $\alpha_t \equiv 1/c$. To guarantee that the iterates produce a nonincreasing sequence of objective function values, we can choose c to satisfy $c \leq \|\mathbf{T}\mathbf{T}^T\|^{-1}\|\mathbf{A}\mathbf{S}\mathbf{S}^T\mathbf{A}^T\|^{-1}$; see [24].

In the experiments described below, we used a data set with dimensions $k = 274$, $m = 224$, $n = 73542$, and there were 1179 and 256 spatial and temporal bases, respectively. A simulated cortical signal was used to generate the data, while matrix $\mathbf{A}$ was derived from a real world experimental setup. White noise was added to the simulated measurements to achieve an SNR (defined as $\|\mathbf{A}\mathbf{X}\|_F^2/E[\mathbf{N}]_F^2$, where $\mathbf{N}$ is additive noise) of 5 dB. The dimension of each coefficient block $\mathbf{\Theta}_{i,j}$ was 3×32. For more detailed information about the experimental set-up, see [7].

We made simple modifications to the EM code to implement other variants of the SpaRSA approach. The changes required to the code were conceptually quite minor; they required only a mechanism for selecting the value of α_t at each iteration (according to formulae such as (21)) and, in the case of monotone methods, increasing this value as needed to obtain a decrease in the objective. The same termination criteria were used for all SpaRSA variants and for EM.

In the cold-start cases, the algorithms were initialized with $\hat{\mathbf{\Theta}}^{(0)} = 0$. In all the SpaRSA variants, the initial value α_0 was set to $2/c$, where c is the constant from (31) that is used in the EM algorithm. (The SpaRSA results are not sensitive to this initial value.)

We used two variants of SpaRSA that were discussed above:

- **SpaRSA:** Choose α_t by the formula (21) at each iteration t;
- **SpaRSA-monotone:** Choose α_t initially by the formula (21) at iteration t, by increase by a factor of 2 as needed to obtain reduction in the objective.

The relative regularization parameter λ was set to various values in the range $(0, 1)$. (For the value $\lambda = 1$, the problem data is such that the solution is $\hat{\mathbf{\Theta}} = 0$.) Convergence testing was performed on only every tenth iteration.

Both MATLAB codes (SpaRSA and EM) were executed on a personal computer with two Intel *Pentium IV* 3 GHz processors and 2GB of memory, running CentOS 4.5 Linux. Table III reports on results obtained by running EM and SpaRSA from the cold start, for various values of λ. Iteration counts and CPU times (in seconds) are shown for the three codes. For SpaRSA-monotone, we also report the total number of function/gradient evaluations, which is generally higher than the iteration count because of the additional evaluations performed during backtracking. The last columns show the final objective value and the number of nonzero blocks. These values differ slightly between codes; we show the output here from the SpaRSA (nonmonotone) runs.

The most noteworthy feature of Table III is the huge improvement in run time of the SpaRSA strategy on this data set over the EM strategy — over two orders of magnitude. In fact, the EM algorithm did not terminate before reaching the upper limit of 10000 function evaluations except in the case $\lambda = 0.7$.

Table IV shows results obtained using a continuation strategy, in which we solve for the largest value $\lambda = 0.7$ (the first value in the table) from a zero initialization, and use the computed solution of each λ value as the starting point for the next value in the table. For the values $\lambda = 0.3$ and $\lambda = 0.2$, the warm start improves the time to solution markedly for the SpaRSA methods. EM also benefits from warm starting, but we do not report the results from this code as the runtimes are still much longer than those of SpaRSA.

TABLE IV
COMPUTATIONAL RESULTS FOR CONTINUATION STRATEGY. TIMES IN SECONDS.

	SpaRSA		SpaRSA-monotone			final	active
λ	its	time	its	evals	time	cost	blocks
0.70	60	18.	30	53	14.	1.5975e-6	2
0.60	40	12.	40	51	14.	1.5440e-6	3
0.50	30	9.	30	40	11.	1.4548e-6	4
0.40	60	17.	50	72	20.	1.3278e-6	4
0.30	70	20.	60	88	24.	1.1621e-6	4
0.25	60	17.	60	94	25.	1.0644e-6	6
0.20	150	43.	90	160	42.	9.5652e-7	9
0.15	110	32.	80	143	37.	8.3749e-7	12
0.10	310	88.	190	359	92.	7.0568e-7	19

V. CONCLUDING REMARKS

In this paper, we have introduced the SpaRSA algorithmic framework for solving large-scale optimization problems involving the sum of a smooth error term and a possibly nonsmooth regularizer. We give experimental evidence that SpaRSA matches the speed of the state-of-the-art method when applied to the $\ell_2 - \ell_1$ problem, and show that SpaRSA can be generalized to other regularizers such as those with group-separable structure. Ongoing work includes a more thorough experimental evaluation involving wider classes of regularizers and other types of data, and theoretical analysis of the convergence properties.

ACKNOWLEDGMENTS

We thank Andrew Bolstad for his help with the description of the MEG brain imaging application and with the computational experiments reported in Section IV-E.

APPENDIX

In this appendix, we present the proof of Theorem 1. We begin by introducing some notation and three technical lemmas which support the main proof. Denoting

$$\begin{aligned} \mathbf{d}^t &= \mathbf{x}^{t+1} - \mathbf{x}^t, \\ \ell(t) &= \arg \max_{i=\max(0,t-M),\ldots,t} \phi(\mathbf{x}^i), \end{aligned} \tag{32}$$

the acceptance condition (22) can be written as

$$\phi(\mathbf{x}^{t+1}) \leq \phi(\mathbf{x}^{\ell(t)}) - \frac{\sigma}{2}\alpha_t \|\mathbf{d}^t\|^2. \tag{33}$$

Our first technical lemma shows that in the vicinity of a noncritical point, and for α_t bounded above, the solution of (5) is a substantial distance away from the current iterate $\mathbf{x}^t$.

Lemma 2: Suppose that $\bar{\mathbf{x}}$ is not critical for (1). Then for any constant $\bar{\alpha} > \alpha_{\min}$, there is $\epsilon(\bar{\alpha}) > 0$ such that for any subsequence $\{\mathbf{x}^{t_j}\}_{j=0,1,2,\ldots}$ with $\lim_{j\to\infty} \mathbf{x}^{t_j} = \bar{\mathbf{x}}$ with $\alpha_{t_j} \in$

TABLE III
COMPUTATIONAL RESULTS FROM $x = 0$ STARTING POINT, FOR VARIOUS VALUES OF λ. TIMES IN SECONDS.. $*$MAXIMUM ITERATION COUNT REACHED PRIOR TO SOLUTION.

	EM		SpaRSA		SpaRSA-monotone			final	active
λ	its	time	its	time	its	evals	time	cost	blocks
0.7	8961	2464.	60	18.	30	53	14.	1.5975e-6	2
0.5	10000*	2749.*	90	26.	80	129	34.	1.4548e-6	4
0.4	10000*	2754.*	90	26.	70	117	31.	1.3278e-6	4
0.3	—		210	60.	140	248	64.	1.1621e-6	4
0.2	—		360	102.	210	369	95.	9.5652e-7	8

$[\alpha_{\min}, \bar{\alpha}]$, we have $\|\mathbf{d}^{t_j}\| = \|\mathbf{x}^{t_j+1} - \mathbf{x}^{t_j}\| > \epsilon(\bar{\alpha})$ for all j sufficiently large.

Proof: Assume for contradiction that for such a sequence, we have $\|\mathbf{d}^{t_j}\| \to 0$, so that $\lim_{j\to\infty} \mathbf{x}^{t_j+1} = \bar{\mathbf{x}}$. By optimality of $\mathbf{x}^{t_j+1} = \mathbf{x}^{t_j} + \mathbf{d}^{t_j}$ in (5), we have

$$0 \in \nabla f(\mathbf{x}^{t_j}) + \alpha_{t_j}\mathbf{d}^{t_j} + \tau\partial c(\mathbf{x}^{t_j+1}).$$

By taking limits as $j \to \infty$, and using outer semicontinuity of ∂c (see [65, Theorem 24.5]) and boundedness of α_{t_j}, we have that (23) holds, contradicting noncriticality of $\bar{\mathbf{x}}$. ∎

The next lemma shows that the acceptance test (22) is satisfied for all sufficiently large values of α_t.

Lemma 3: Let $\sigma \in (0, 1)$ be given. Then there is a constant $\tilde{\alpha} > 0$ such that for any sequence $\{\mathbf{x}^{t_j}\}_{j=0,1,2,\ldots}$, the acceptance condition (22) is satisfied whenever $\alpha_{t_j} \geq \tilde{\alpha}$.

Proof: We show that in fact

$$\phi(\mathbf{x}^{t_j+1}) \leq \phi(\mathbf{x}^{t_j}) - \frac{\sigma}{2}\alpha_{t_j}\|\mathbf{d}^{t_j}\|^2,$$

which implies (22) for $t = t_j$. Denoting by γ the Lipschitz constant for ∇f, we have

$$\begin{aligned}
&\phi(\mathbf{x}^{t_j+1}) - \phi(\mathbf{x}^{t_j}) = \\
&= f(\mathbf{x}^{t_j+1}) + \tau c(\mathbf{x}^{t_j+1}) - f(\mathbf{x}^{t_j}) - \tau c(\mathbf{x}^{t_j}) \\
&\leq \nabla f(\mathbf{x}^{t_j})^T\mathbf{d}^{t_j} + \gamma\|\mathbf{d}^{t_j}\|^2 + \tau c(\mathbf{x}^{t_j+1}) - \tau c(\mathbf{x}^{t_j}) \\
&\leq \left(\gamma - \frac{1}{2}\alpha_{t_j}\right)\|\mathbf{d}^{t_j}\|^2,
\end{aligned}$$

where the last inequality follows from the fact that $\mathbf{x}^{t_j+1}$ achieves a better objective value in (5) than $\mathbf{z} = \mathbf{x}^{t_j}$. The result then follows provided that

$$\gamma - \frac{1}{2}\alpha_{t_j} \leq -\frac{\sigma}{2}\alpha_{t_j},$$

which is in turn satisfied whenever $\alpha_{t_j} \geq \tilde{\alpha}$, where $\tilde{\alpha} := 2\gamma/(1-\sigma)$. ∎

Our final technical lemma shows that the step lengths obtained by solving (5) approach zero, and that the full sequence of objective function values has a limit.

Lemma 4: The sequence $\{\mathbf{x}^t\}$ generated by Algorithm SpaRSA with acceptance test (22) has $\lim_{t\to\infty} \mathbf{d}^t = 0$. Moreover there exists a number $\bar{\phi}$ such that $\lim_{t\to\infty} \phi(\mathbf{x}^t) = \bar{\phi}$.

Proof: Recalling the notation (32), note first that the sequence $\{\phi(\mathbf{x}^{\ell(t)})\}_{t=0,1,2,\ldots}$ is monotonically decreasing, because from (32) and (33) we have

$$\begin{aligned}
\phi(\mathbf{x}^{\ell(t+1)}) &= \max_{j=0,1,\ldots,\min(M,t+1)} \phi(\mathbf{x}^{t+1-j}) \\
&= \max\left\{\max_{j=1,\ldots,\min(M,t+1)} \phi(\mathbf{x}^{t+1-j}), \phi(\mathbf{x}^{t+1})\right\} \\
&\leq \max\left\{\phi(\mathbf{x}^{\ell(t)}), \phi(\mathbf{x}^{\ell(t)}) - \frac{\sigma}{2}\alpha_t\|\mathbf{d}^t\|^2\right\} \\
&= \phi(\mathbf{x}^{\ell(t)}).
\end{aligned}$$

Therefore, since ϕ is bounded below, there exists $\bar{\phi}$ such that

$$\lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)}) = \bar{\phi}. \tag{34}$$

By applying (33) with t replaced by $\ell(t) - 1$, we obtain

$$\phi(\mathbf{x}^{\ell(t)}) \leq \phi(\mathbf{x}^{\ell(\ell(t)-1)}) - \frac{\sigma}{2}\alpha_{\ell(t)-1}\|\mathbf{d}^{\ell(t)-1}\|^2;$$

by rearranging this expression and using (34), we obtain

$$\lim_{t\to\infty} \alpha_{\ell(t)-1}\|\mathbf{d}^{\ell(t)-1}\|^2 = 0,$$

which, since $\alpha_r \geq \alpha_{\min}$ for all r, implies that

$$\lim_{t\to\infty} \mathbf{d}^{\ell(t)-1} = 0. \tag{35}$$

We have from (34) and (35) that

$$\begin{aligned}
\bar{\phi} &= \lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)}) \\
&= \lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-1} + \mathbf{d}^{\ell(t)-1}) \\
&= \lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-1}).
\end{aligned} \tag{36}$$

We will now prove, by induction, that the following limits are satisfied for all $j \geq 1$:

$$\begin{aligned}
&\lim_{t\to\infty} \mathbf{d}^{\ell(t)-j} = 0, \\
&\lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-j}) = \bar{\phi}.
\end{aligned} \tag{37}$$

We have already shown in (35) and (36) that the results holds for $j = 1$; we now need to show that if they hold for j, then they also hold $j+1$. From (33) with t replaced by $\ell(t)-j-1$, we have

$$\phi(\mathbf{x}^{\ell(t)-j}) \leq \phi(\mathbf{x}^{\ell(\ell(t)-j-1)}) - \frac{\sigma}{2}\alpha_{\ell(t)-j-1}\|\mathbf{d}^{\ell(t)-j-1}\|^2.$$

(We have assumed that t is large enough to make the indices $\ell(t) - j - 1$ nonnegative.) By rearranging this expression and using $\alpha_r \geq \alpha_{\min}$ for all r, we obtain

$$\|\mathbf{d}^{\ell(t)-j-1}\|^2 \leq \frac{2}{\sigma\alpha_{\min}}\left[\phi(\mathbf{x}^{\ell(\ell(t)-j-1)}) - \phi(\mathbf{x}^{\ell(t)-j})\right].$$

By letting $t \to \infty$, and using the inductive hypothesis along with (34), we have that the right-hand side of this expression

approaches zero, and hence $\lim_{t\to\infty} \mathbf{d}^{\ell(t)-(j+1)} = 0$, proving the inductive step for the first limit in (37). The second limit in (37) follows immediately, since

$$\begin{aligned}\lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-(j+1)}) &= \lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-(j+1)} + \mathbf{d}^{\ell(t)-(j+1)}) \\ &= \lim_{t\to\infty} \phi(\mathbf{x}^{\ell(t)-j}) \\ &= \bar{\phi}.\end{aligned}$$

To complete our proof that $\lim_{t\to\infty} \mathbf{d}^t = 0$, we note that $\ell(t)$ is one of the indices $t-M, t-M+1, \ldots, t$. Hence, we can write $t-M-1 = \ell(t)-j$ for some $j = 1, 2, \ldots, M+1$. Thus from the first limit in (37), we have $\lim_{t\to\infty} \mathbf{d}^t = \lim_{t\to\infty} \mathbf{d}^{t-M-1} = 0$. For the limit of function values, we have that, for all t,

$$\mathbf{x}^{\ell(t)} = \mathbf{x}^{t-M-1} + \sum_{j=1}^{\ell(t)-(t-M-1)} \mathbf{d}^{\ell(t)-j},$$

thus $\lim_{t\to\infty}(\mathbf{x}^{\ell(t)} - \mathbf{x}^{t-M-1}) = 0$. It follows from continuity of ϕ and the second limit in (37) that $\lim_{t\to\infty} \phi(\mathbf{x}^t) = \bar{\phi}$. ■

We now prove Theorem 1.

Proof: (Theorem 1) Suppose (for contradiction) that $\bar{\mathbf{x}}$ is an accumulation point that is not critical. Let $\{t_j\}_{j=0,1,2,\ldots}$ be the subsequence of indices such that $\lim_{j\to\infty} \mathbf{x}^{t_j} = \bar{\mathbf{x}}$. If the parameter sequence $\{\alpha_{t_j}\}$ were bounded, we would have from Lemma 2 that $\|\mathbf{d}^{t_j}\| = \|\mathbf{x}^{t_j+1} - \mathbf{x}^{t_j}\| \geq \epsilon$ for some $\epsilon > 0$ and all j sufficiently large. This contradicts Lemma 4, so we must have that $\{\alpha_{t_j}\}$ is unbounded. In fact we can assume without loss of generality that $\{\alpha_{t_j}\}$ increases monotonically to ∞ and that $\alpha_{t_j} \geq \eta \max(\alpha_{\max}, \tilde{\alpha})$ for all j. For this to be true, the value $\alpha = \alpha_{t_j}/\eta$ must have been tried at iteration t_j and must have failed the acceptance test (22). But Lemma 3 assures us that (22) must have been satisfied for this value of α, a further contradiction.

We conclude that no noncritical point can be an accumulation point, proving the theorem. ■

REFERENCES

[1] J. Barzilai, J. Borwein, "Two point step size gradient methods," *IMA Journal of Numerical Analysis,* vol. 8, pp. 141–148, 1988.

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

[3] E. Birgin, J. Martinez, M. Raydan, "Nonmonotone spectral projected gradient methods on convex sets," *SIAM Journal on Optimization,* vol. 10, pp. 1196–1211, 2000.

[4] T. Blumensath and M. Davies. "Gradient pursuits," *IEEE Transactions on Signal Processing*, 2008 (to appear). Available at `www.see.ed.ac.uk/~tblumens`

[5] A. Bolstad, B. Van Veen, R. Nowak, "Space-time sparsity regularization for the magnetoencephalography inverse problem", *Proceedings of the IEEE International Conference on Biomedical Imaging*, Arlington, VA, 2007.

[6] A. Bolstad, B. Van Veen, R. Nowak, R. Wakai, "An expectation-maximization algorithm for space-time sparsity regularization of the MEG inverse problem", *Proceedings of the International Conference on Biomagnetism*, Vancouver, BC, Canada, 2006.

[7] A. Bolstad, B. Van Veen, R. Nowak "Magneto-/electroencephalography with Space-Time Sparse Priors", *IEEE Statistics and Signal Processing Workshop*, Madison, WI, USA, 2007.

[8] S. Bourguignon, H. Carfantan, and J. Idier. "A sparsity-based method for the estimation of spectral lines from irregularly sampled data", *IEEE Journal of Selected Topics in Signal Processing,* vol. 1, pp. 575–585, 2007.

[9] E. Candès, J. Romberg and T. Tao. "Stable signal recovery from incomplete and inaccurate information," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1233, 2005.

[10] E. Candès, J. Romberg, and T. Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory,* vol. 52, pp. 489–509, 2006.

[11] E. Candès and T. Tao, "The Dantzig selector: statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, pp. 2313–2351, 2007.

[12] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematics Imaging and Vision*, vol. 20, pp. 89–97, 2004.

[13] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," in *Mathematical Models of Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras (Eds), Springer Verlag, 2005.

[14] C. Chaux, P. Combettes, J.-C. Pesquet, V. Wajs, "A variational formulation for frame-based inverse problems," *Inverse Problems*, vol. 23, pp. 1495–1518, 2007.

[15] S. Chen, D. Donoho, and M. Saunders. "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computation*, vol. 20, pp. 33–61, 1998.

[16] J. Claerbout and F. Muir. "Robust modelling of erratic data," *Geophysics*, vol. 38, pp. 826–844, 1973.

[17] P. Combettes, V. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Journal on Multiscale Modeling & Simulation*, vol. 4, pp. 1168–1200, 2005.

[18] Y.-H. Dai, R. Fletcher. "Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming," *Numerische Mathematik*, vol. 100, pp. 21–47, 2005.

[19] Y.-H. Dai, W. Hager, K. Schittkowski, H. Zhang, "The cyclic Barzilai-Borwein method for unconstrained optimization," *IMA Journal of Numerical Analysis*, vol. 26, pp. 604–627, 2006.

[20] J. Darbon, M. Sigelle, "Image restoration with discrete constrained total variation; part I: fast and exact optimization", *Journal of Mathematical Imaging and Vision*, vol 26, pp. 261–276, 2006.

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

[22] I. Daubechies, M. Fornasier, I. Loris, "Accelerated projected gradient method for linear inverse problems with sparsity constraints," *Journal of Fourier Analysis and Applications*, 2008 (to appear). Available at `http://arxiv.org/abs/0706.4297`.

[23] G. Davis, S. Mallat, M. Avellaneda, "Greedy adaptive approximation," *Journal of Constructive Approximation,* vol. 12, pp. 57–98, 1997.

[24] A. Dempster, N. Laird, and D. Rubin. "Maximum likelihood estimation from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[25] P. Djurić. "A model selection rule for sinusoids in white Gaussian noise," *IEEE Transactions on Signal Processing,* vol. 44, pp. 1744–1751, 1996.

[26] D. Donoho. "De-noising by soft thresholding," *IEEE Transactions on Information Theory*, vol. 41, pp. 6–18, 1995.

[27] D. Donoho. "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, 2006.

[28] D. Donoho, M. Elad, and V. Temlyakov. "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, pp. 6–18, 2006.

[29] D. Donoho and Y. Tsaig. "Fast solution of L1-norm minimization problems when the solution may be sparse," Technical Report 2006-18, Department of Statistics, Stanford University, 2006.

[30] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, submitted, 2007. Available at `http://stat.stanford.edu/~idrori/StOMP.pdf`.

[31] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the L1-ball for learning in high dimensions," *International Conference on Machine Learning – ICML'2008,* Helsinki, 2008.

[32] M. Elad, "Why simple shrinkage is still relevant for redundant representations?", *IEEE Transactions on Information Theory*, vol. 52, pp. 5559–5569, 2006.

[33] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. "Least Angle Regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[34] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representations", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition – CVPR'2006*, New York, 2006.

[35] M. Figueiredo and R. Nowak. "Wavelet-based image estimation: an empirical Bayes approach using Jeffreys' noninformative prior," *IEEE Transactions on Image Processing*, vol. 10, pp. 1322–1331, 2001.

[36] M. Figueiredo, R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, pp. 906–916, 2003.

[37] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration" *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[38] M. Figueiredo, J. Bioucas-Dias, J. Oliveira, and R. Nowak, "On total-variation denoising: A new majorization-minimization algorithm and an experimental comparison with wavalet denoising," *IEEE International Conference on Image Processing – ICIP'06*, 2006.

[39] M. Figueiredo, R. Nowak, S. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, pp. 586–598, 2007.

[40] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. "Pathwise coordinate optimization", *Annals of Applied Statistics,* vol. 1, pp. 302–332, 2007.

[41] J. Fuchs, "More on sparse representations in arbitrary bases, *IEEE Transactions on Information Theory,* vol. 50, pp. 1341-1344, 2004.

[42] J. Fuchs, "Convergence of a sparse representations algorithm applicable to real or complex data," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, pp. 598–605, 2007.

[43] H. Gao, "Wavelet shrinkage denoising using the non-negative garrote," *Journal of Computational and Graphical Statistics*, vol. 7, pp. 469–488, 1998.

[44] H. Gao and A. Bruce, "Waveshrink with firm shrinkage," *Statistica Sinica*, vol. 7, pp. 855–874, 1997.

[45] D. Goldfarb and W. Yin, "Parametric maximum flow algorithmsfor fast total variation minimization," Technical Report TR07-09, Department of Computational and Applied Mathematics, Rice University, 2007.

[46] G. Golub and C. Van Loan. *Matrix Computations*, 3rd ed., Johns Hopkins University Press, 1996.

[47] I. Gorodnitsky and B. Rao. "Sparse signal reconstruction from limited data using FOCUSS: a recursive weighted norm minimization algorithm," *IEEE Transactions on Signal Processing,* vol. 45, pp. 600-616, 1997.

[48] R. Griesse and D. Lorenz, "A semismooth Newton method for Tikhonov functionals with sparsity constraints," *Inverse Problems*, vol. 24, no. 3, 2008.

[49] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM Journal on Numerical Analysis* 23 (1986), pp. 707–716.

[50] L. Grippo, M. Sciandrone, "Nonmonotone globalization techniques for the Barzilai-Borwein method," *Computational Optimization and Applications*, vol. 32, pp. 143–169, 2002.

[51] T. Hale, W. Yin, Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing." TR07-07, Department of Computational and Applied Mathematics, Rice University, 2007.

[52] J. Haupt and R. Nowak. "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, pp. 4036–4048, 2006.

[53] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky. " An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, pp. 606–617, 2007.

[54] Y. Kim, J. Kim, Y. Kim, "Blockwise sparse regression", *Statistica Sinica*, vol. 16, pp. 375–390, 2006.

[55] S. Levy and P. Fullagar. "Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution," *Geophysics*, vol. 46, pp. 1235–1243, 1981.

[56] D. Malioutov, M. Çetin, and A. Willsky. "Homotopy continuation for sparse signal representation," *Proceedings of the IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 733–736, Philadelphia, PA, 2005.

[57] D. Malioutov, M. Cetin, A. Willsky, "Sparse signal reconstruction perspective for source localization with sensor arrays", *IEEE Transactions on Signal Processing*, vol. 53, pp. 3010–3022, 2005.

[58] L. Meier, S. van de Geer, P. Buhlmann, "The group LASSO for logistic regression",*Journal of the Royal Statistical Society B*, vol. 70, pp. 53–71, 2008.

[59] A. Miller, *Subset Selection in Regression.* Chapman and Hall, London, 2002.

[60] P. Moulin and J. Liu. "Analysis of multiresolution image denoising schemes using generalized-Gaussian and complexity priors," *IEEE Transactions on Information Theory,* vol. 45, pp. 909–919, 1999.

[61] Y. Nesterov, "Gradient methods for minimizing composite objective function" CORE Discussion Paper 2007/76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2007.

[62] J. Nocedal, S. J. Wright. *Numerical Optimization*, 2nd Edition, Springer, 2006.

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

[64] S. Osher, L. Rudin, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D.*, vol. 60, pp. 259–268, 1992.

[65] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[66] F. Santosa and W. Symes. "Linear invesion of band-limited reflection histograms," *SIAM Journal of Scientific and Statistical Computing,* vol. 7, pp. 1307–1330, 1986.

[67] M. A. Saunders. "PDCO: Primal-dual interior-point method for convex objectives," Systems Optimization Laboratory, Stanford University, 2002. Available at `www.stanford.edu/group/SOL/`

[68] T. Serafini, G. Zanghirati, L. Zanni. "Gradient projection methods for large quadratic programs and applications in training support vector machines," *Optimization Methods and Software*, vol. 20, pp. 353–378, 2004.

[69] H. Taylor, S. Bank, J. McCoy. "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, pp. 39–52, 1979.

[70] R. Tibshirani. "Regression shrinkage and selection via the lasso," *Journal Royal Statostical Society B*, vol. 58, pp. 267-288, 1996.

[71] J. Tropp. "Just relax: Convex programming methods for identifying sparse signals," *IEEE Transactions on Information Theory,* vol. 51, pp. 1030–1051, 2006.

[72] J. Tropp. "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory,* vol. 50, pp. 2231–2242, 2004.

[73] P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization", *Journal of Optimization Theory and Applications,* vol. 109, pp. 475-494, 2001.

[74] B. Turlach, W. N. Venables, and S. J. Wright. "Simultaneous variable selection," *Technometrics*, vol. 27, pp. 349-363, 2005.

[75] E. van den Berg and M. P. Friedlander, "In Pursuit of a root," Technical Report TR-2007-19, Department of Computer Science, University of British Columbia, June 2007.

[76] S. Weisberg. *Applied Linear Regression*. John Wiley & Sons, New York, 1980.

[77] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, pp. 143–168, 2008.

[78] M. Yuan, Y. Lin, "Model selection and estimation in regression with grouped variables", *Journal of the Royal Statistical Society B*, vol. 68, pp. 49-67, 2006.

[79] P. Zhao, G. Rocha, B. Yu, "Grouped and hierarchical model selection through composite absolute penalties", TR 703, Statistics Department, University of California - Berkeley, 2007.

[80] P. Zhao, B. Yu, "Boosted LASSO", Technical Report, Statistics Department, University of California - Berkeley, 2004.

[81] C. Zhu, "Stabe recovery of sparse signals via regularized minimization", *IEEE Transactions on Information Theory,* vol. 54, pp. 3364–3367, 2008.