A Primal Dual Active Set Algorithm with Continuation for Compressed Sensing

Qibin Fan, Yuling Jiao, Xiliang Lu

Abstract—The success of compressed sensing relies essentially on the ability to efficiently find an approximately sparse solution to an under-determined linear system. In this paper, we developed an efficient algorithm for the sparsity promoting $\ell_1$-regularized least squares problem by coupling the primal dual active set strategy with a continuation technique (on the regularization parameter). In the active set strategy, we first determine the active set from primal and dual variables, and then update the primal and dual variables by solving a low-dimensional least square problem on the active set, which makes the algorithm very efficient. The continuation technique globalizes the convergence of the algorithm, with provable global convergence under restricted isometry property (RIP). Further, we adopt two alternative methods, i.e., a modified discrepancy principle and a Bayesian information criterion, to choose the regularization parameter. Numerical experiments indicate that our algorithm is very competitive with state-of-the-art algorithms in terms of accuracy and efficiency.

Index Terms—compressive sensing, $\ell_1$-regularization, primal dual active set method, continuation, modified discrepancy principle, Bayesian information criterion.

I. INTRODUCTION

COMPRESSIVE sensing (CS) has recently emerged as a promising approach for acquiring (approximately) sparse signals. An important problem in CS is to find the sparsest solution of the following under-determined linear system [1]–[3]

$$\Psi x = y, \quad (1)$$

where $\Psi \in \mathbb{R}^{n \times p}$ is the sampling matrix with $n \ll p$, $x$ is a sparse signal, $y$ is the measurement, which may contain noise. It can be equivalently written as an optimization problem

$$\min_{x \in \mathbb{R}^p} \left\| x \right\|_0, \quad \text{subject to} \quad \left\| \Psi x - y \right\|_2 \leq \epsilon, \quad (2)$$

where $\left\| x \right\|_0$ denotes the the number of nonzero entries in the vector $x$ and $\epsilon$ is the noise level. Due to the nonsmooth and nonconvex structure of problem (2), it is very challenging to find the sparsest solution. Now it is widely accepted that the $\ell_1$ convex relaxation can provide a satisfactory approximate solution, if the solution $x$ and the sampling matrix $\Psi$ satisfies certain conditions.

There are three different versions of $\ell_1$ convex relaxation that have received a lot of attentions. They are Basis Pursuit Denoising (BPDN) [4]:

$$\min_{x \in \mathbb{R}^p} \left\| x \right\|_1, \quad \text{subject to} \quad \left\| \Psi x - y \right\|_2 \leq \epsilon, \quad (3)$$

the $\ell_1$-regularized least squares problem [4]:

$$\min_{x \in \mathbb{R}^p} \frac{1}{2} \left\| \Psi x - y \right\|_2^2 + \lambda \left\| x \right\|_1, \quad (4)$$

and the least absolute shrinkage and selection operator (LASSO) [5] model:

$$\min_{x \in \mathbb{R}^p} \left\| \Psi x - y \right\|_2, \quad \text{subject to} \quad \left\| x \right\|_1 \leq \tau. \quad (5)$$

Where $\lambda$ and $\tau$ are regularized parameter and tuning parameter, respectively. It was shown in [6] if these parameters are chosen properly, problems (3) - (5) have the same minimizer.

In this paper, we are interested in the fast solution of the $\ell_1$-regularized least squares model (4).

Over the last few years, a large number of algorithms have been developed for problems (3) - (5). We will list only a few exemplary methods here, and refer to the review [7]–[9] for a comprehensive overview. Gradient type methods, e.g., gradient projection sparse reconstruction [10], sparse reconstruction via separable approximation [11], spectral gradient projection [6], fixed point iteration with continuation strategy [12], [13], iterative shrinkage/thresholding algorithm [14], [15] and their accelerated extension [16], [17], [18], are extremely popular. Other classical methods, e.g., homotopy method [19]–[21], alternating direction method of multipliers [22], iteratively reweighted least square method [23], have also received revived interest in solving $\ell_1$ minimization problems.

These algorithms can have only sublinear or linear convergence rate. Therefore, it is of immense interest to develop Newton type algorithms that enjoy a (locally) superlinear convergence rate. For an invertible matrix $\Psi$, the primal dual active set (PDAS) method (also known as semismooth Newton method), has been studied in [24]–[26]. This idea can be extended to the CS setting to solve problem (4). Theoretically, it enjoys a locally superlinear convergence.

However, in Newton type algorithm, a good initial guess is very important for the successful application of the PDAS method. Unlike gradient based algorithms, the PDAS method does not have a monotonic decreasing property for the cost functional. Therefore without a good initial guess the algorithm may not converge. Meanwhile, in the model (4), the regularization parameter $\lambda$ balances the sparsity of the solution and the fidelity of the measurements. And its proper choice plays an essential role for getting a satisfactory reconstruction.

In this article we propose a simple but efficient technique to find a good initial guess by combining the continuation strategy with primal dual active set algorithm. Moreover, equipped with a proper stop rule, the regularization parameter can be chosen automatically without much adding work. To be
precise, the $l_1$-regularized minimization problems are solved with warm start on a predefined decreasing sequence $\{\lambda_i\}$, i.e., the solution $x_i(\lambda_i)$ to $\lambda_i$-problem is chosen as the initial guess for $\lambda_{i+1}$-problem solved by PDAS. It needs only a few (Newton) steps since $x_i(\lambda_i)$ provides a good initial guess.

The main contributions of this paper are twofold. First, we derive a local one step convergence result for $\lambda_{i+1}$-problem which improves the well known local superlinear convergence of PDAS [24]. More importantly, we prove the global convergence for the primal dual active set algorithm with continuation (PDASC) under the standard restricted isometry property (RIP) assumption on the matrix $\Psi$ in the noise-free case. On the other hand, when the measurement involves noise, we adopt the parameter selection rule based on either a modified discrepancy principle or Bayesian information criterion. One can use this rule to select a suitable regularization parameter $\lambda$ and solution $x(\lambda)$ during the continuation process with nearly no adding effort.

The rest of the paper is organized as follows. In section 2 we introduce the mathematical background, PDAS algorithm and continuation technique, discuss their convergence properties and the regularization parameter selection rule. In section 3, several numerical examples are presented to illustrate the efficiency and accuracy PDASC algorithm, by comparing with several state-of-the-art sparse reconstruction algorithms. The technical proofs are in the appendices.

II. PDASC ALGORITHM

A. Notations

Given a vector $x = (x_1, x_2, ..., x_p)^t \in \mathbb{R}^p$, we denote by $\|x\|_q = \left(\sum_{i=1}^p |x_i|^q \right)^{\frac{1}{q}}$ with $q \in [1, \infty)$ and $\|x\|_{\infty} = \max_{1 \leq i \leq p} |x_i|$. Further, $\Psi^t$ and $\|\Psi\|$ denote the transpose and 2-norm of the matrix $\Psi$, respectively. The matrix $\Psi$ is assumed to be columnwise normalized, i.e., $\|\Psi\|_2 = 1$ for $i = 1, ..., p$.

The notation 1 (or 0) refers to a column vector with all entries equal to 1 (or 0). For any set

$$A \subseteq S \triangleq \{1, 2, ..., p\}$$

of size $|A|$, $x_A \in \mathbb{R}^{|A|}(\Psi_A \in \mathbb{R}^{n \times |A|})$ is the subvector (submatrix) whose entries (columns) are listed in $A$.

We denote by $\Gamma_0(\mathbb{R}^p)$ the set of all proper lower semicontinuous convex functions on $\mathbb{R}^p$. The subdifferential of any $f \in \Gamma_0(\mathbb{R}^p)$ is a set-valued mapping defined by

$$\partial f(z) := \{w \in \mathbb{R}^p : f(v) \geq f(z) + \langle w, v - z \rangle, \forall v \in \mathbb{R}^p\}.$$ 

The subdifferential of $f = \|x\|_1$ is the pointwise set-value sign function $\text{Sign}(x)$ [27], i.e.,

$$z \in \text{Sign}(x) \iff z_i = 1, \quad x_i > 0,$$

$$z \in \text{Sign}(x) \iff z_i = -1, \quad x_i < 0,$$

$$z \in \text{Sign}(x) \iff z_i = 0, \quad \{1, -1\}, \quad x_i = 0. \quad (6)$$

The classical Fermat’s rule for proper lower semicontinuous convex functions [28] asserts

$$0 \in \partial f(z^*) \iff z^* \text{ is a minimizer of } f. \quad (7)$$

For a given $f \in \Gamma_0(\mathbb{R}^p)$, the proximal operator $\text{Prox}_f$ is defined by $\text{Prox}_f(z) := \text{argmin}_{x \in \mathbb{R}^p} \{\frac{1}{2}\|x - z\|^2 + f(x)\}$. Then there holds [29]

$$w \in \partial f(z) \iff z = \text{Prox}_f(z + w). \quad (8)$$

The proximal operator of $\|\cdot\|_1$ is given by the pointwise soft-thresholding operator [27]

$$\text{Prox}_{\lambda\|\cdot\|_1}(z) = T_\lambda(x), \quad (9)$$

where

$$z = T_\lambda(x) \iff z_i = \max\{|x_i| - \lambda, 0\}\text{sign}(x_i). \quad (10)$$

B. Motivation and PDAS Algorithm

Now we characterize the minimizer of (4) by its KKT system (c.f. [12], which motivates the PDAS algorithm; see also Appendix A for a short proof, which is included for completeness.

**Theorem 1:** If $x^* \in \mathbb{R}^p$ is a minimizer of (4), then there exists a $d^* \in \mathbb{R}^p$ such that the KKT system holds:

$$\Psi^t \Psi x^* + d^* = \Psi^t y, \quad (11)$$

$$x^* = T_\lambda(x^* + d^*). \quad (12)$$

Conversely, if $x^* \in \mathbb{R}^p$ and $d^* \in \mathbb{R}^p$ satisfying (11) and (12), then $x^*$ is a minimizer of (4).

Let $x^*$ and $d^*$ be the optimal primal and dual variables. Clearly, it follows from (12) that

$$x^*_i > 0 \iff x^*_i + d^*_i > \lambda, \quad x^*_i < 0 \iff x^*_i + d^*_i < -\lambda.$$

Hence, one can use the information from both primal and dual variables, rather than the primal variable alone, to determine the nonzero components of $x^*_i$ (which is called active set). This motivates us to define the active and inactive sets by:

$$A^+_\lambda = \{i \in S : x^*_i + d^*_i > \lambda\},$$

$$A^-_\lambda = \{i \in S : x^*_i + d^*_i < -\lambda\},$$

$$A_\lambda = A^+_\lambda \cup A^-_\lambda, \quad I_\lambda = A^+_\lambda.$$ \quad (13)

Then the KKT system (11)-(12) can be reformulated. First, by (12) and the soft thresholding operator (10), we deduce

$$x^*_{I_\lambda} = 0_{I_\lambda}. \quad (14)$$

Meanwhile, the proof of Theorem 1 implies $d^* \in \lambda\partial \|\cdot\|_1(x^*)$. Then by (6), we get $d^*_A^+ = \lambda I A^+_\lambda$, and $d^*_A^- = -\lambda I A^-_\lambda$, i.e.,

$$d_A = \lambda [I A^+ + I A^-]. \quad (15)$$

Upon relabeling, (11) can be equivalently written as

$$\begin{pmatrix} \Psi^t \Psi_{A^+} & \Psi^t \Psi_{A^-} \\ \Psi^t \Psi_{A^+} & \Psi^t \Psi_{A^-} \end{pmatrix} \begin{pmatrix} x_{A^+} \\ x_{A^-} \end{pmatrix} + \begin{pmatrix} d_{A^+} \\ d_{A^-} \end{pmatrix} = \begin{pmatrix} \Psi^t \Psi y \\ \Psi^t \Psi y \end{pmatrix}, \quad (16)$$

which, in view of the relations (14) and (15), can be further rewritten as

$$\Psi^t \Psi_{A^+} x_{A^+} = \Psi^t \Psi y - d_{A^+}, \quad (17)$$

$$d_{A^-} = \Psi^t \Psi y - \Psi^t \Psi A^+_\lambda x_{A^+}. \quad (18)$$

Hence, if the active set $A_\lambda$ is known, then the optimal solution $(x^*, d^*)$ follows directly from (14), (15), (17) and (18). This motivates a PDAS algorithm. Suppose $x^k \in \mathbb{R}^p$
and \( d^k \in \mathbb{R}^p \) are approximations to \( x^* \) and \( d^* \). Similar to (13), we define the active and inactive sets by

\[
\begin{align*}
A^+_k &= \{ i \in S : x^+_k + d^+_k > \lambda \}, \\
A^-_k &= \{ i \in S : x^-_k + d^-_k < -\lambda \}, \\
A_k &= A^+_k \cup A^-_k, \\
I_k &= A^+_k \cup A^-_k + 1.
\end{align*}
\]

(19)

Then hopefully, the active set \( A_k \) and inactive set \( I_k \) are also good approximations of \( A^* \) and \( I^* \), respectively. Now by repeating the arguments leading to (14), (15), (17) and (18), we update \( x^{k+1} \) and \( d^{k+1} \) by the following systems:

\[
\begin{align*}
x^{k+1} &= 0_{I_k+1}, \\
d^{k+1} &= \lambda [I_{A^+_k+1}, -I_{A^-_k+1}]^t, \\
\Psi_A^t \Psi_A x^{k+1} &= \Psi_A y - d^{k+1}, \\
d^{k+1} &= \Psi_{I_k+1} y - \Psi_{I_k+1} x^{k+1} \Psi_A x^{k+1}.
\end{align*}
\]

(20), (21) and (23) involve only matrix-vector multiplications, and thus they are computationally efficient. The well-posedness of the system depends on the solvability of (22), which in turn depends on the property of the submatrix \( \Psi_{A_k \cup I_k} \). In the compressible sensing problem, the active set \( A^* \) is often small. Then if \( A_k \) is an approximation of \( A^* \), it is also small and \( \Psi_{A_k \cup I_k} \) is likely to be a full-column rank matrix. We will discuss the well-posedness in subsection E below. Now we summarize the PDAS method in Algorithm 1.

**Algorithm 1 PDAS**

1: Input: initial guess \((x^0, d^0), \lambda \), and \( J \).
2: for \( k = 0, 1, 2, 3, \cdots \) do
3: \( A^+_k \) and \( I^+_k \) by (19).
4: \( x^k = 0_{I_k+1} \).
5: \( d^k = \lambda [I_{A^+_k+1}, -I_{A^-_k+1}]^t \).
6: \( x^{k+1} = (\Psi_{A_k \cup I_k} \Psi_{A_k \cup I_k}^{-1})(\Psi_{A_k \cup I_k} y - d^{k+1}) \).
7: \( d^{k+1} = \Psi_{I_k \cup I_k} y - \Psi_{I_k \cup I_k} x^{k+1} \).
8: Check stopping rule (either \( A^+_k = A^+_k \) or \( k+1 \geq J \)).
9: end for
10: Output approximation \((x^{k+1}, d^{k+1})\).

**C. Complexity analysis**

First, we consider the number of floating point operations per iteration. Clearly it takes \( O(p) \) flops to finish steps 3 - 5 in the PDAS. In step 6, forming the matrix \( \Psi_{A_k \cup I_k} \) explicitly takes \( O(n |A_k \cup I_k|^2) \) flops (the cost of forming the right hand side is negligible since \( \Psi^t y \) can be precomputed and retrieved efficiently). The Cholesky factorization costs \( O(|A_k \cup I_k|^3) \) flops and the back-substitution needs \( O(|A_k \cup I_k|^2) \) flops. Hence step 6 takes \( O(|A_k \cup I_k|^2 \max(n, |A_k \cup I_k|)) \) flops. At step 7, two matrix-vector products cost at most \( O(np) \) flops. So, the the overall cost of the PDAS per iteration is \( O(max(|A_k \cup I_k|^3, pn, |A_k \cup I_k|^2)n) \).

The next issue is the number of iterations. Since the PDAS is equivalent to the semi-smooth Newton method [24], [34], a local superlinear convergence is guaranteed. The numerical experiments in section 3 also indicate that it converges within a few iterations. So with a good initial guess, the overall cost of the PDAS is also \( O(max(|A_k \cup I_k|^3, pn, |A_k \cup I_k|^2)n) \).

If the sought-for solution is sufficiently sparse, i.e.,\( |A_k \cup I_k| < \min(n, \sqrt{p}) \), the cost of per PDAS iteration is \( O(np) \) by applying Cholesky up/down-date [31]. To be precise, we downdate by removing the columns in \( \Psi_{A_k} \) but not in \( \Psi_{A_k \cup I_k} \) at the cost of \( O(|A_k \cup I_k| \min(|A_k \cup I_k|, n) \log n) \) flops, and update by appending the columns in \( \Psi_{A_k \cup I_k} \) but not in \( \Psi_{A_k} \) in \( O(|A_k \cup I_k| \min(|A_k \cup I_k|, n) \log n) \) flops. Then the cost of Cholesky factorization of \( \Psi^t \Psi_{A_k \cup I_k} \) is \( O(|A_k \cup I_k| \min(|A_k \cup I_k|, n) \log n) \). Further, with warm starting, the difference between \( A_k \) and \( A_k \cup I_k \) is small. Hence, \( \min(|A_k \cup I_k|, n) \) is not large, and \( \min(|A_k \cup I_k|, n) \) usually holds.

**Remark 2.1:** Algorithm 1 requires the explicit form of \( \Psi \). Often the signals are sparse or compressible only in a certain basis. Then the sensing matrix \( \Psi \) is the product of a (random) sampling matrix and the transform matrix, i.e., \( \Psi \) is only given implicitly. One can avoid the explicit expression of \( \Psi \) by solving the linear system at step 6 iteratively, e.g., with conjugate gradient method (CG). It involves only matrix-vector multiplications, which can often been carried out efficiently for structured \( \Psi \). Only a few CG iterations are needed due to the well-conditionedness of the system.

**D. Continuation technique**

In view of the equivalence of the PDAS and the semismooth Newton method [24], a good initial guess is essential to its success. For nonsmooth optimization problems, there are several ways to globalize the Newton method, including squared smoothing with line search [32] or path-following with model function detection [33]. Due to the special structure of CS problems, we adopt a continuation technique. Specifically, we consider a decreasing sequence of parameter \{\( \lambda_s \)\}, and apply Algorithm 1 to \( \lambda_s \)-problem with the initial guess from the solution of the \( \lambda_s \)-problem. Summarizing the idea leads to Algorithm 2.

**Algorithm 2 PDASC**

1: Input: \( \lambda_0 \geq ||\Psi^t y||_{\infty} \), \( x(\lambda_0) = 0_S, d(\lambda_0) = \Psi^t y, \rho \in (0, 1) \).
2: for \( s = 1, 2, 3, \cdots \) do
3: set \( \lambda_s = \lambda_0 \rho^s \) and \( (x^0, d^0) = (x(\lambda_{s-1}), d(\lambda_{s-1})) \).
4: Find \( x(\lambda_s) \) and \( d(\lambda_s) \) by Algorithm 1.
5: Check stopping rule.
6: end for
7: Output: approximation \( (x(\lambda_s), d(\lambda_s)) \).

**E. Convergence analysis**

We first consider the convergence of Algorithm 1. The local superlinear convergence of the PDAS can be obtained by reformulating it in the semismooth Newton framework [24],
Then on the active set, there holds $\|x^\delta - \tilde{x}^\delta\| \leq \rho(1 + \delta_k)\|x\|_2$ for all $x$ with $\|x\|_0 \leq k$.

**Assumption 1:** $\Psi$ satisfies RIP of order $T + 1$, and the RIP constant $\delta \triangleq \delta_{T + 1} \leq \sqrt{T}$.

**Theorem 3:** Let Assumption 1 be fulfilled. With the choice $\rho = \frac{1}{2}$ in Algorithm 2, and $J \geq T$ in Algorithm 1, Algorithm 2 is well-defined. Further, for sufficiently large $s$, the support of $x(\lambda_s)$ is $A^1$, and $\lim_{s \to \infty} x(\lambda_s) = x^1$.

**Remark 2.3:**
1. Theorem 3 considers only the noise-free case. In the noisy case, if the noise level is small, the algorithm is still well defined when equipped with a suitable stopping rule.
2. $J$ is to ensure that Algorithm 1 stops with a finite iteration. In practice, it is not necessary to be large.
3. Assumption 1 (with slightly different constant) has been used in the proof of the convergence for orthogonal matching pursuit algorithm (OMP) [35].

### F. Selection of regularization parameter $\lambda$

Now we discuss the stopping rule at line 5 of Algorithm 2 and the choice of regularization parameter $\lambda$.

If the noise level $\epsilon$ is known, the discrepancy principle ($\|\Psi x - y\|_2 \leq \epsilon$) is widely applied to choose a suitable regularization parameter in inverse problem [36]. However, for CS problems, it tends to choose a solution with a very large active set; see the numerical examples in Section 3. This is attributed to the fact that the $\ell_1$-regularized model may lead to a biased solution [37]. More precisely, suppose that the true active set $A^1$ were found, and thus primal and dual variables satisfies

$$d_{A^1} = \Psi_{A^1}^T (y - \Psi x), \quad |d_{A^1}| = \lambda |1_{A^1}|.$$

This implies that the residual term $\Psi x - y$ may not be small and hence the discrepancy principle may not satisfied. Meanwhile, let the oracle solution be

$$x^\delta_{A^1} = \tilde{\Psi}_{A^1}^T y = (\Psi_{A^1}^T \Psi_{A^1})^{-1} \tilde{\Psi}_{A^1}^T y.$$

Then on the active set, there holds $x^\delta_{A^1} + (\Psi_{A^1}^T \Psi_{A^1})^{-1} d_{A^1} = x^\delta_{A^1}$. Hence, $x^\delta_{A^1} + (\Psi_{A^1}^T \Psi_{A^1})^{-1} d_{A^1}$ is a better approximation to the true solution. This motivates us to propose a modified discrepancy principle (MDP) for the stopping rule and selecting the regularization parameter. Specifically, let the active set of $x(\lambda_s)$ in Algorithm 2 be $A_s$. Algorithm 2 stops when

$$\|\Psi_{A_s}(x(\lambda_s)_{A_s} + (\Psi_{A_s}^T \Psi_{A_s})^{-1} d(\lambda_s)_{A_s}) - y\|_2 \leq \epsilon,$$

where $\epsilon$ is the noise level, and accordingly the approximate solution is given by

$$x_{A_s} = x(\lambda_s)_{A_s} + (\Psi_{A_s}^T \Psi_{A_s})^{-1} d(\lambda_s)_{A_s}, \quad x_{I_s} = 0_{I_s}.$$

The above equation is a debias step, see also [10] for the similar debias postprocessing. One should be noticed that after this debias postprocess, the solution obtained may not be the solution to (4), but be more close to solution of $\ell_2$-minimization problem. This debias postprocess will only been done when the modified discrepancy principle (MDP) is satisfied.

If the noise level is unknown, we choose the stopping criterion at line 5 of Algorithm 2 as the size of the active set, e.g., $\|x(\lambda_s)\|_0 \geq \eta n$ for $\eta \in [0.5, 1]$. To choose a proper regularization parameter $\lambda$, we employ Bayesian information criterion (BIC), which is a data driven method and widely used in statistics due to its model selection consistency [38], [39]. BIC chooses $\lambda$ by:

$$\min_{\lambda \in \Lambda} \left\{ \text{BIC}(\lambda) := \frac{1}{2} \|\Psi x_\lambda - y\|_2^2 + \frac{\ln n}{n} \Lambda f_\lambda \right\}, \tag{24}$$

where $x_\lambda$ is the solution of (4), $\Lambda$ is a subset of $\{0, +\infty\}$, and $df_\lambda$ represents the degree of freedom of $x_\lambda$ that can be chosen as $\|x_\lambda\|_0$ [40]. Due to the complex structure of the BIC functional, it is nontrivial to find its minimizer over the whole positive real line. Instead, a practical way is to find the minimizer over the finite candidate set $\Lambda = \{\lambda_s\}_s$ which will be specified in the next section in numerical tests.

### III. Numerical Examples

Now we present numerical examples to show the efficiency and accuracy of Algorithm 2 (PDASC). First, we give the implementation details, e.g., the generation of simulation data, parameter setting for the algorithm. Then we check the efficiency of regularization parameter choice strategy: for both MDP and BIC based parameter choice rules. Later on our method is also compared with several state-of-the-art algorithms for both CPU time and reconstruction error.

#### A. Implementation Setting

The signals $x^1$ are chosen as $T$-sparse with a dynamic range

$$\text{Dyna} := \frac{\max\{|x^1_i| : x^1_i \neq 0\}}{\min\{|x^1_i| : x^1_i \neq 0\}}.$$

They are generated following [18].

The sensing matrix $\Psi$ of size $n \times p$ is chosen to be either random Gaussian matrix, or random Bernoulli matrix, or partial discrete cosine transform (DCT) matrix. The observation vector $y$ is given by $y = \Psi x^1 + \eta$, where $\eta$ is the Gaussian noise vector whose entries are i.i.d. $\sim N(0, \sigma)$.
One needs the following algorithm parameters: initial regularization parameter $\lambda_0$; decreasing factor $\rho$; maximal iteration number $J$, and noise level $\epsilon$. The noise level is chosen as $\epsilon = \|\eta\|$. The maximal iteration number $J$ is not sensitive to the algorithm (due to the locally superlinear or one step convergence property of PDAS), one can choose it as $J = 1$. To determine the initial regularization parameter $\lambda_0$ and decreasing factor $\rho$, we pickup an interval $[\lambda_{\min}, \lambda_{\max}]$ which contains the target the regularization parameter. Then an equal-distributed partition on log-scale is employed to divide this interval into $N$-subintervals. Clearly larger $N$ implies larger $\rho$. For simplicity, let $\lambda_{\max} = \|\Psi^T \eta\|_\infty$, $\lambda_{\min} = 1e-10 \lambda_{\max}$ and $N = 100$.

When the sensing matrix $\Psi$ is random Bernoulli matrix or random Gaussian matrix, the matrix $\Psi^T \Psi$ is saved in advance (not be included in CPU time), and the linear equation in line 6 of Algorithm 1 is solved by Cholesky factorization. But when $\Psi$ is a partial discrete cosine transform matrix, we do not have the explicit form of $\Psi^T \Psi$. The linear equation in line 6 of Algorithm 1 is solved by conjugated gradient (CG) method initialized with the projection of the previous solution onto the current active set. We set the number of CG iteration as 2 in all the simulations below.

### B. Check regularization parameter selection rules

We will check the ability of proposed regularization selection rules. The three rules are compared in Table I, they are modified discrepancy principle (MDP), Bayesian information criterion (BIC), and standard discrepancy principle (DP). We consider nine different cases where the sensing matrix $\Psi$ is chosen as $512 \times 2048$ partial DCT matrix, $256 \times 1024$ random Gaussian matrix, and $200 \times 1000$ random Bernoulli matrix, respectively. For each type of sensing matrix $\Psi$, we consider three different noise level and different sparsity level. The details is shown in Table I.

The first two columns of Table I are different problem setting and different parameter selection rules. The third and forth columns are the CPU time (in seconds) and relatively $\ell_2$ error. The fifth and sixth columns are information of active set. Column five and six are the size of $A \setminus A^1$ and $A^1 \setminus A$, respectively, where $A$ and $A^1$ be the numerical active set and true active set. The last column is the selected regularization parameter $\hat{\lambda}$. In some cases DP may fail and we use $F$ to indicate it.

It should be noticed that the regularization parameter from MDP is much larger than the ones from BIC or DP. The reason is that Algorithm 2 stops immediately when active set $\hat{A}$ contains the true active set $A^1$ due to the debias step. Hence the debias postprocess makes Algorithm 2 terminate earlier and make larger $\lambda$ to be selected. One can find in Table I that when noise level and sparsity level are small, three methods all work well. When the noise level and sparsity level are relatively large, DP may fail, MDP and BIC still work. In most cases, MDP takes less CPU time and chooses a (more accurate) active set, but it requires the information of noise level. In later numerical tests, if the noise level is known, we can use either MDP and BIC to find a solution, otherwise only BIC is available.

### C. Comparison with other algorithms

We compare our algorithm with the several state-of-the-art algorithms for solving (4). The parameters in these algorithms are the default values as their online packages, except for the stopping criterion which will be discussed later.

Gradient projections for sparse reconstruction (GPSR) [10] uses Barzilai-Borwein rule to choose step length. The MATLAB code is available at http://www.lx.it.pt/mtf/GPSR/.

The Matlab code for sparse reconstruction by separable approximation (SpaRSA) [11] is available at http://www.lx.it.pt/mtf/SpaRSA/.

The package of fixed point continuation (FPC) [12] and its modified version (FPC-AS) [13] are available at http://www.caam.rice.edu/~optimization/L1/.

For all these algorithms, a regularization parameter is needed. Since the solution by MDP is slightly different from the solution to (4), we use BIC to pickup a regularization parameter and use it in other algorithms.

### TABLE I: Comparison for choosing regularization parameter

| setting | method | time(s) | error | active set | $\lambda$ |
|---------|--------|---------|-------|------------|-----------|
| Partial DCT | MDP | 0.30 | 4.66e-5 | 0 | 1.80e-1 |
| $\sigma = 1e-4$ | BIC | 0.49 | 3.83e-4 | 0 | 3.79e-4 |
| $|A| = 32$ | DP | 0.39 | 1.37e-4 | 106 | 1.05e-4 |
| Partial DCT | MDP | 0.23 | 4.57e-6 | 0 | 1.62e-1 |
| $\sigma = 1e-2$ | BIC | 0.25 | 3.96e-6 | 10 | 3.57e-2 |
| $|A| = 64$ | DP | 0.26 | 1.50e-4 | 173 | 1.13e-3 |
| Partial DCT | MDP | 0.22 | 5.48e-2 | 50 | 1.79e-1 |
| $\sigma = 5e-2$ | BIC | 0.19 | 9.29e-2 | 85 | 7.92e-2 |
| $|A| = 80$ | DP | 0.39 | F | F | F |

### TABLE II: Random Bernoulli matrix

| method | Time | $\ell_2$ RE | $\ell_\infty$ AE | $\ell_2$ dRE | $\ell_\infty$ dAE |
|--------|------|-------------|-----------------|--------------|-----------------|
| PDASC-11(MDP) | 1.85 | 4.80e-6 | 3.10e-3 | 4.80e-6 | 3.10e-3 |
| PDASC-11(BIC) | 3.58 | 8.90e-5 | 3.96e-2 | 5.14e-6 | 3.28e-3 |
| GPSR-bb | 9.82 | 1.45e-4 | 7.18e-2 | 7.07e-6 | 3.65e-3 |
| SpaRSA | 10.1 | 1.05e-4 | 5.12e-2 | 5.64e-6 | 3.44e-3 |
| FPC | 42.9 | 2.69e-4 | 1.27e-1 | 2.54e-4 | 1.15e-1 |
| FPC-AS | 5.57 | 9.56e-5 | 4.32e-2 | 3.83e-6 | 2.81e-3 |

$n = 2048$, $p = 32768$, $T = 128$, Dyna=1e3, $\sigma = 1e-3$.

As was pointed out in [18], to compare different algorithms, one needs a fair stopping criterion. We setup the stop condition for other algorithm as follows. Firstly we use BIC to get a regularization parameter $\lambda$ and a solution $x(\lambda)$. Then the
stopping rule for other $\ell_1$ solvers is either their default stop criterions or the following condition is fulfilled:

$$\frac{1}{2}\|\Psi y - x\|_2^2 + \lambda\|x\|_1 \leq \frac{1}{2}\|\Psi x(\hat{\lambda}) - y\|_2^2 + \lambda\|x(\hat{\lambda})\|_1.$$

The first group experiments are to recover three different $T$-sparse signal with $T = 128, 256, 1024$, which are sampled by random Bernoulli matrix with size $4096 \times 16384$, random Gaussian matrix with size $2048 \times 32768$, and partial DCT matrix with size $16384 \times 65536$, respectively. The dynamic range of these tests are $1e3, 1e4, 1e2$, respectively. The noise $\sigma$ is chosen as $1e-3, 1e-2, 1e-2$, respectively. The averaged results based on 10 independent replications (CPU times, $\ell_2$ relative errors ($\ell_2$ RE), $\ell_\infty$ absolute errors ($\ell_\infty$ AE), $\ell_2$ relative errors after debias ($\ell_2$ dRE) and $\ell_\infty$ absolute errors after debias ($\ell_\infty$ dAE) are reported in Tables II - IV.

**TABLE III: Random Gaussian matrix**

| method     | Time | $\ell_2$ RE | $\ell_\infty$ AE | $\ell_2$ dRE | $\ell_\infty$ dAE |
|------------|------|-------------|------------------|--------------|------------------|
| PDASC-l1(MDP) | 3.02 | 4.6e-6 | 3.3e-2 | 4.6e-6 | 3.3e-2 |
| PDASC-l1(BIC) | 4.53 | 1.5e-3 | 2.6e-2 | 1.2e-5 | 5.4e-2 |
| GPSR-bb | 6.37 | 1.8e-3 | 8.7e-2 | 1.8e-5 | 6.9e-2 |
| SpaRSA | 10.2 | 5.6e-5 | 9.2e-2 | 1.5e-5 | 6.8e-2 |
| FPC | 25.4 | 3.5e-5 | 9.2e-2 | 1.9e-5 | 9.1e-2 |
| FPC-AS | 6.19 | 1.8e-3 | 7.7e-2 | 1.5e-2 | 6.8e-2 |

$n = 4096, p = 16384, T = 256,$ Dyna=$1e4, \sigma = 1e-2$.

**TABLE IV: partial DCT matrix**

| method     | Time | $\ell_2$ RE | $\ell_\infty$ AE | $\ell_2$ dRE | $\ell_\infty$ dAE |
|------------|------|-------------|------------------|--------------|------------------|
| PDASC-l1(MDP) | 1.56 | 6.5e-4 | 0.08 | 6.5e-4 | 0.08 |
| PDASC-l1(BIC) | 1.02 | 2.0e-3 | 0.13 | 1.9e-3 | 0.11 |
| GPSR-bb | 0.87 | 2.1e-3 | 0.14 | 2.0e-3 | 0.11 |
| SpaRSA | 1.14 | 2.0e-3 | 0.13 | 1.9e-3 | 0.11 |
| FPC | 0.76 | 2.1e-3 | 0.15 | 2.1e-3 | 0.12 |
| FPC-AS | 0.68 | 2.0e-3 | 0.12 | 1.6e-3 | 0.10 |

$n = 16384, p = 65536, T = 1024,$ Dyna=$1e2, \sigma = 1e-2$.

In Table II - IV, PDASC with MDP (the noise level is supposed to known) or BIC are compared with other four algorithms. The first two columns are method and CPU time (in seconds), and last four columns are errors of the solutions. Columns three and four are standard relatively $\ell_2$ error and absolute $\ell_\infty$ error. The last two columns are the $\ell_2$ and $\ell_\infty$ after a debias postprocess. It is observed that Algorithm 2 is very competitive to other state-of-art algorithms in both accuracy and CPU time. However, the regularization parameter is not necessarily known in advance for PDASC which may make PDASC a good candidate for large scale real data. If the sensing matrix is random Bernoulli or random Gaussian, PDASC with MDP is fastest, and when $\Psi$ is partial DCT matrix PDASC with MDP is a bit slower. This fact is due to that we apply different solvers for the linear system in step 6 of Algorithm 1, i.e., Cholesky factorization for previous two cases (the explicit form of $\Psi^\dagger \Psi$ is needed) and CG for the last case, respectively.

Next group of numerical examples reconstruct a one dimensional signal and a benchmark MRI image. Both of them are compressible under a Haar wavelet basis. Therefor, the observation data can be chosen as the wavelet coefficients sampled by the product of a partial FFT matrix and inverse Haar wavelet transform. Similarly one needs a regularization parameter for other state-of-the-art algorithms. Same as before, we first run Algorithm 2 with BIC to get a regularization parameter $\hat{\lambda}$, and use it for other $\ell_1$ solvers. In these two examples we assume the noise level is not known (this is the case for most real data) and we use PDASC with BIC to compare with other solver by CPU time and PSNR value. The stopping rule for other algorithms are the same as before. The results are reported in Table V, VI and Figure 1.

**TABLE V: One dimensional signal**

| method     | CPU time | PSNR |
|------------|----------|------|
| PDASC-l1 | 0.50 | 54   |
| GPSR-bb | 0.62 | 54   |
| SpaRSA | 0.70 | 54   |
| FPC | 0.42 | 54   |
| FPC-AS | 0.70 | 54   |

$n = 665, p = 1024, T = 247, \sigma = 1e-4, \hat{\lambda} = 7.42e-4$.

**Fig. 1: Reconstruction signal and their PSNR values**

For the one dimensional signal, the sampling matrix $\Psi$ with size $665 \times 1024$ is the compound of a partial FFT and a inverse wavelet transform, and the signal under wavelet transformation has 247 nonzero entries. The sampling matrix $\Psi$ for two dimensional MRI image is the compound of a partial FFT and an inverse wavelet transform with size $2133 \times 4096$. The image under wavelet transformation has 792 nonzero entries. The numerical results also demonstrate that the proposed PDASC is very competitive in terms of efficiency and accuracy, but without a priori knowledge of regularization parameter.

**TABLE VI: Two dimensional imagine**

| method     | CPU time | PSNR |
|------------|----------|------|
| PDASC-l1 | 0.52 | 66   |
| GPSR-bb | 0.76 | 65   |
| SpaRSA | 0.86 | 66   |
| FPC | 0.92 | 65   |
| FPC-AS | 1.75 | 66   |

$n = 2133, p = 4096, T = 702, \sigma = 1e-4, \hat{\lambda} = 5.35e-4$. 
IV. CONCLUSION

A primal dual active set with continuation algorithm together with suitable regularization parameter choice rules has been proposed to solve $\ell_1$-regularized least squares problem. We derived the local one step convergence of PDAS and established the global convergence of PDASC. Numerical experiments verified the algorithm PDASC is very competitive and remains unclear. Second, BIC is very promising data driven regularization parameters selection rule, but its efficient implementation is still challenging. Last, adaptation of the algorithm to more complicated scenarios, such as severely ill-posed inverse problems, is also of immense practical interest.

APPENDIX A

**Proof of Theorem 1**

*Proof:* Let $x^* \in \mathbb{R}^p$ be a minimizer of (4), then by (7) we have

$$0 \in \Psi'(\Psi x^* - y + \lambda d) \cdot \|1(x^*)\).$$
(25)

Therefore, there exists $d^* \in \lambda \partial \|1(x^*)\) such that $0 = \Psi'(\Psi x^* - y + d^*)$. Now (8) and (9) imply,

$$d^* \in \lambda \partial \|1(x^*)\) \iff x^* = \text{Prox}_{\lambda \|1\}}(x^* + d^*) = T_\lambda(x^* + d^*)$$

Conversely, suppose (11) and (12) hold. By (8) and (11), we obtain that $d^* \in \lambda \|1(x^*)\). Substitute to (11) we have $0 \in \Psi'(\Psi x^* - y + \lambda \|1(x^*)\). By Fermat’s rule (7), we conclude that $x^*$ is a minimizer of (4).

**APPENDIX B**

**Proof of Theorem 2**

*Proof:* Let $J^* = \{i : |x^*_i + d^*_i| \neq \lambda\}$, and

$$\theta = \min_{i \in J^*} \|x^*_i + d^*_i\| - \lambda > 0.$$
for any $i \in I^f$ we have 
$$
|d_i| \leq \delta(\|x_i^f\| + \|d_A\| + \|x_A^f - x_A - d_A\|)
$$
$$
\leq \frac{\delta}{1 - \delta}\|x_i^f\| + \frac{\delta}{1 - \delta}\|d_A\|.
$$

Let $i_A \in \arg \max_{i \in I} \|x_i^f\|$. Clearly $i_A \in B$, and hence 
$$
|d_{i_A}| \geq \|x_i^f\|_{i_A} - \delta(\|x_i^f\| + \|d_A\| + \|x_A^f - x_A - d_A\|)
$$
$$
\geq \|x_i^f\|_{i_A} - \frac{\delta}{1 - \delta}\|x_i^f\| - \frac{\delta}{1 - \delta}\|d_A\|.
$$

By the trivial estimates $\|x_i^f\| \leq \sqrt{\|B\|}\|x_i^f\|$, $\|d_A\| = \sqrt{\|A\|\lambda}$, and the implication $\delta \leq \frac{\sqrt{\|B\|}}{\sqrt{T+1}} \leq \frac{1}{4}$, we deduce 
$$
\|x_i^f + d_A - x_A^f\| \leq \frac{1}{4}\|x_i^f\|_{i_A} + \frac{1}{4}\lambda,
$$
(26)

$$
|d_i| \leq \frac{1}{4}\|x_i^f\|_{i_A} + \frac{1}{4}\lambda, \forall i \in I^f.
$$
(27)

Further, for any given $\lambda > 0$ and $m > 0$, we define the set 
$$
J_{\lambda,m} = \{i : \|x_i^f\| \geq m\lambda\}.
$$
(29)

The proof of Theorem 3 is based on the following claim for any iteration of Algorithm 1.

Claim 1: Let $m = 2$ or $3$.

a. If $J_{\lambda,2} \subseteq A_k \subseteq A^f$, then $J_{\lambda,2} \subseteq A_{k+1} \subseteq A^f$.

b. If $J_{\lambda,3} \subseteq A_k \subseteq A^f$, we have either $J_{\lambda,2} \subseteq A_k$ or 
$$
\max\{\|x_i^f\| : i \in I_k\} > \max\{\|x_i^f\| : i \in I_{k+1}\}.
$$

Proof: By the assumption $J_{\lambda,m} \subseteq A_k \subseteq A^f$, we have $\|x_i^f\| < m\lambda$. Combining estimates (26)-(28) yields for $m = 2, 3$ 
$$
\|x_i^f + d_i^k\| \geq \frac{3}{4}\|x_i^f\| - \frac{1}{4}\lambda \geq \frac{3m - 1}{4}\lambda > \lambda, \forall i \in J_{\lambda,m},
$$
$$
|d_i| \leq \frac{1}{4}\|x_i^f\|_{i_A} + \frac{1}{4}\lambda < \frac{m + 1}{4}\lambda < \lambda, \forall i \in I^f,
$$
which implies that $J_{\lambda,m} \subseteq A_{k+1} \subseteq A^f$. Now we assume $J_{\lambda,3} \subseteq A_k \subseteq A^f$ and $J_{\lambda,2} \subseteq A_k$. Then for any $i_A \in J_{\lambda,3} \setminus J_{\lambda,2}$, $\|x_i^f\|_{i_A} \in (2\lambda, 3\lambda)$. Consider any $i \in A_k$ such that 
$$
\|x_i^f\| \geq \|x_i^f\|_{i_A}.
$$
we have 
$$
\|x_i^f + d_i^k\| \geq \frac{3}{4}\|x_i^f\|_{i_A} - \frac{1}{4}\lambda \geq \lambda \Rightarrow i \in A_{k+1}.
$$
For any $i_A$, we also have 
$$
|d_i| \geq \frac{1}{4}\|x_i^f\|_{i_A} - \frac{1}{4}\lambda > \lambda \Rightarrow i_A \in A_{k+1}.
$$

Therefore 
$$
\max\{\|x_i^f\| : i \in I_k\} > \max\{\|x_i^f\| : i \in I_{k+1}\}.
$$

Now we state the proof of Theorem 3.

Proof: For any given $\lambda$, let Algorithm 1 take $k_* \lambda$-steps to stop and denote the active set during the PDAS iteration (cf. Algorithm 1) by $A_{k_*}$ for $k \leq k_*$, and 
$$
A_{0,\lambda} = \{i : |x_i^{k_*} + d_i^{k_*}| > \lambda\}.
$$

By construction (cf. Algorithm 2), we have $(x^{k_*},d^{k_*}) = (x(\lambda), d(\lambda))$, and it is the initial guess for $\lambda_{k+1}$-problem. We shall prove $A_{k_*} \subseteq A^f$ by mathematical induction and hence also the well-posedness of the algorithm. To this end, we need the following claim:

Claim 2: For any $s \geq 0$, we have $J_{\lambda_{s+1},3} \subseteq A_{1,s} \subseteq A^f$ and $J_{\lambda_{s+1},2} \subseteq A_{0,s} \subseteq A^f$.

Step 1. For any $s \geq 0$, if $J_{\lambda_{s+1},3} \subseteq A_{1,s} \subseteq A^f$, then by Claim 1, we have $J_{\lambda_{s+1},3} \subseteq A_{k_*} \subseteq A^f$ for any $k \leq k_*$. When Algorithm 1 stops, it is either $A_{k_*} = A_{k_*+1,s}$ or $k_* = J \geq T$. By Claim 1, in both cases, we have $J_{\lambda_{s+1},2} \subseteq A_{0,s} \subseteq A^f$.

Step 2. Consider the case $s = 1$. Upon noting $\lambda_0 > \|\Psi f\|_\infty$, there holds $J_{1,3} = \emptyset$. To see this, we let $\|x_i^f\| = \max_{j=1,\ldots,T} |x_j^f|$, then 
$$
\|\Psi f\| \geq \|x_i^f\| - \delta\sqrt{T}\|x_i^f\| \geq \frac{\delta}{2}\|x_i^f\| \Rightarrow \|x_i^f\| < 3\lambda_1 \Rightarrow J_{1,3} = \emptyset.
$$
By mathematical induction, noting the relations $\lambda_{s+1} = \frac{2}{\lambda_s}$ and $J_{1,3} = A^f$, and hence Algorithm 1 converges in one step and the support of $x(\lambda_s)$ is $A^f$. The last assertion follows 
$$
\|x^f - x(\lambda_s)\|_{A^f} = (\Psi A^f\Psi A^f)^{-1}\|d(\lambda_s)\|_{A^f}
$$
and 
$$
\|d(\lambda_s)\|_{A^f} = \lambda_s.
$$

ACKNOWLEDGMENT

The work of Q. Fan was partially supported by National Science Foundation of China No. 61179039 and the work of X. Lu is partially supported by National Science Foundation of China No. 11101316 and No. 91230108. The authors would like to thank the anonymous referees for their constructive comments. The authors would also like to thank Dr. Bangti Jin for useful discussions.

REFERENCES

[1] E. Candés, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, 2006.

[2] D. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] J. Tropp, “Just relax: Convex programming methods for identifying sparse signals in noise,” IEEE Transactions on Information Theory, vol. 52, no. 1, pp. 1030–1051, 2006.

[4] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” SIAM Journal on Scientific Computing, vol. 20, no. 1, pp. 33–61, 1998.

[5] R. Tibshirani, “Regression shrinkage and selection via the lasso,” Journal of the Royal Statistical Society. Series B (Methodological), pp. 267–288, 1996.

[6] E. Van Den Berg and M. Friedlander, “Probing the pareto frontier for basis pursuit solutions,” SIAM Journal on Scientific Computing, vol. 31, no. 2, pp. 890–912, 2008.

[7] J. Tropp and S. Wright, “Computational methods for sparse solution of linear inverse problems,” Proceedings of the IEEE, vol. 98, no. 6, pp. 948–958, 2010.

[8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, “Optimization with sparsity-inducing penalties,” Foundations and Trends® in Signal Processing, vol. 5, no. 1-2, 2012.

[9] P. Combettes and J. Pesquet, “Proximal splitting methods for signal processing,” Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212, 2011.

[10] M. Figueiredo, R. Nowak, and S. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 586–597, 2007.

[11] S. Wright, R. Nowak, and M. Figueiredo, “Sparse reconstruction by separable approximation,” IEEE Transactions on Signal Processing, vol. 57, no. 7, pp. 2479–2493, 2009.

[12] E. Hale, W. Yin, and Y. Zhang, “Fixed-point continuation for $\ell_1$-minimization: Methodology and convergence,” SIAM Journal on Optimization, vol. 19, no. 3, pp. 1107–1130, 2008.
[13] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, “A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation,” SIAM Journal on Scientific Computing, vol. 32, no. 4, pp. 1832–1857, 2010.

[14] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” Communications on Pure and Applied Mathematics, vol. 57, no. 11, pp. 1413–1457, 2004.

[15] P. Combettes and V. Wajs, “Signal recovery by proximal forward-backward splitting,” Multiscale modeling & simulation, vol. 4, no. 4, pp. 1168–1200, 2005.

[16] Y. Nesterov, “Smooth minimization of non-smooth functions,” Mathematical programming, vol. 103, no. 1, pp. 127–152, 2005.

[17] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183–202, 2009.

[18] S. Becker, J. Bobin, and E. Candés, “NESTA: a fast and accurate first-order method for sparse recovery,” SIAM Journal on Imaging Sciences, vol. 4, no. 1, pp. 1–39, 2011.

[19] M. Osborne, B. Presnell, and B. Turlach, “A new approach to variable selection in least squares problems,” IMA Journal of Numerical Analysis, vol. 20, no. 3, pp. 389–403, 2000.

[20] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” The Annals of Statistics, vol. 32, no. 2, pp. 407–499, 2004.

[21] D. Donoho, “The fast solution of $l_1$-norm minimization problems when the solution may be sparse,” Department of Statistics, Stanford University, 2006.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.

[23] I. Daubechies, R. DeVore, M. Fornasier, and C. Gntrk, “Iteratively reweighted least squares minimization for sparse recovery,” Communications on Pure and Applied Mathematics, vol. 63, no. 1, pp. 1–38, 2009.

[24] M. Hintermüller, K. Ito, and K. Kunisch, “The primal-dual active set strategy as a semismooth newton method,” SIAM Journal on Optimization, vol. 13, no. 3, pp. 865–888, 2002.

[25] R. Griesse and D. Lorenz, “A semismooth newton method for tikhonov functionals with sparsity constraints,” Inverse Problem, vol. 24, no. 3, pp. 035007, 19 pp.

[26] B. Jin, D. Lorenz, and S. Schiffler, “Elastic-net regularization: error estimates and active set methods,” Inverse Problem, vol. 25, no. 11, pp. 115022, 26 pp., 2009.

[27] D. Donoho and I. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” J. Amer. Statist. Assoc., vol. 90, no. 432, pp. 1200–1224, 1995.

[28] R. Rockafellar, Convex analysis. Princeton university press, 1996, vol. 28.

[29] C. Micchelli, L. Shen, and X. Y., “Proximity algorithms for image models: denoising,” Inverse Problems, vol. 27, no. 05, pp. 045009, 30 pp., 2011.

[30] K. Ito and K. Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications. SIAM, Philadelphia, 2008.

[31] G. Golub and C. Van Loan, Matrix computations. Johns Hopkins University Press, 1996, vol. 3.

[32] D. Sun and L. Qi, “Solving variational inequality problems via smoothing-non-smooth reformulations,” Journal of computational and applied mathematics, vol. 129, no. 1, pp. 37–62, 2001.

[33] M. Hintermüller and K. Kunisch, “Path-following methods for a class of constrained minimization problems in function space,” SIAM Journal on Optimization, vol. 17, no. 1, pp. 159–187, 2006.

[34] M. Grasmair, O. Scherzer, and M. Haltmeier, “Necessary and sufficient conditions for linear convergence of $l_1$-regularization,” Communications on Pure and Applied Mathematics, vol. 64, no. 2, pp. 161–182, 2011.

[35] M. A. Davenport and M. B. Wakin, “Analysis of orthogonal matching pursuit using the restricted isometry property,” Information Theory, IEEE Transactions on, vol. 56, no. 9, pp. 4395–4401, 2010.

[36] H. W. Engl, M. Hanke, and A. Neubauer, Regularization of inverse problems. Springer, 1996, vol. 375.

[37] C.-H. Zhang and J. Huang, “The sparsity and bias of the lasso selection in high-dimensional linear regression,” The Annals of Statistics, vol. 36, no. 4, pp. 1567–1594, 2008.

[38] S. Konishi and G. Kitagawa, Information criteria and statistical modeling. Springer, 2007.