IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 8, 2018, accepted December 8, 2018, date of publication December 12, 2018,
date of current version January 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886528

A Unified Framework for Sparse Relaxed Regularized Regression: SR3

PENG ZHENG[1], (Student Member, IEEE), TRAVIS ASKHAM[1], STEVEN L. BRUNTON[2], (Member, IEEE), J. NATHAN KUTZ[1], (Member, IEEE), AND ALEKSANDR Y. ARAVKIN[1], (Member, IEEE)

[1]Department of Applied Mathematics, University of Washington, Seattle, WA 98195-3925, USA
[2]Department of Mechanical Engineering, University of Washington, Seattle, WA 98195-2600, USA

Corresponding author: Aleksandr Y. Aravkin (saravkin@uw.edu)

The work of P. Zheng and A. Y. Aravkin was supported by the Washington Research Foundation Data Science Professorship. The work of T. Askham and J. N. Kutz was supported by the Air Force Office of Scientific Research under Grant FA9550-17-1-0329. The work of S. L. Brunton was supported by the Army Research Office through the Young Investigator Program under Grant W911NF-17-1-0422.

ABSTRACT Regularized regression problems are ubiquitous in statistical modeling, signal processing, and machine learning. Sparse regression, in particular, has been instrumental in scientific model discovery, including compressed sensing applications, variable selection, and high-dimensional analysis. We propose a broad framework for sparse relaxed regularized regression, called SR3. The key idea is to solve a *relaxation* of the regularized problem, which has three advantages over the state-of-the-art: 1) solutions of the relaxed problem are superior with respect to errors, false positives, and conditioning; 2) relaxation allows extremely fast algorithms for both convex and nonconvex formulations; and 3) the methods apply to composite regularizers, essential for total variation (TV) as well as sparsity-promoting formulations using tight frames. We demonstrate the advantages of SR3 (computational efficiency, higher accuracy, faster convergence rates, and greater flexibility) across a range of regularized regression problems with synthetic and real data, including applications in compressed sensing, LASSO, matrix completion, TV regularization, and group sparsity. Following standards of reproducible research, we also provide a companion MATLAB package that implements these examples.

INDEX TERMS Nonconvex optimization, sparse regression, compressed sensing, LASSO, total variation regularization, matrix completion.

I. INTRODUCTION

Regression is a cornerstone of data science. In the age of big data, optimization algorithms are largely focused on regression problems in machine learning and AI. As data volumes increase, algorithms must be fast, scalable, and robust to low-fidelity measurements (missing data, noise, and outliers). Regularization, which includes priors and constraints, is essential for the recovery of interpretable solutions in high-dimensional and ill-posed settings. Sparsity-promoting regression is one such fundamental technique, that enforces solution parsimony by balancing model error with complexity. Despite tremendous methodological progress over the last 80 years, many difficulties remain, including (i) restrictive theoretical conditions for practical performance, (ii) the lack of fast solvers for large scale and ill-conditioned problems, (iii) practical difficulties with nonconvex implementations, and (iv) high-fidelity requirements on data. To overcome these difficulties, we propose a broadly applicable method, *sparse relaxed regularized regression* (SR3), based on a relaxation reformulation of *any* regularized regression problem. We demonstrate that SR3 is fast, scalable, robust to noisy and missing data, and flexible enough to apply broadly to regularized regression problems, ranging from the ubiquitous LASSO and compressed sensing (CS), to composite regularizers such as the total variation (TV) regularization, and even to nonconvex regularizers, including ℓ_0 and rank. SR3 improves on the state-of-the-art in all of these applications, both in terms of computational speed and performance. Moreover, SR3 is flexible and simple to implement. A companion open source package implements a range of examples using SR3.

The origins of regression extend back more than two centuries to the pioneering mathematical contributions of Legendre [37] and Gauss [30], [31], who were interested in determining the orbits of celestial bodies. The invention of the digital electronic computer in the mid 20th century greatly increased interest in regression methods, as computations became faster and larger problems from a variety of fields became tractable. It was recognized early on that many regression problems are ill-posed in nature, either being under-determined, with an infinite set of candidate solutions, or otherwise sensitive to perturbations in the observations, often due to some redundancy in the set of possible models. Tikhonov [50] was the first to systematically study the use of regularizers to achieve stable and unique numerical solutions of such ill-posed problems. The regularized linear least squares problem is given by

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2 + \lambda R(\mathbf{C}\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the unknown signal, $\mathbf{A} \in \mathbb{R}^{m\times d}$ is the linear data-generating model for the observations $\boldsymbol{b} \in \mathbb{R}^m$, $\mathbf{C} \in \mathbb{R}^{n\times d}$ is a linear map, $R(\cdot)$ is any regularizer, and λ parametrizes the strength of the regularization. Tikhonov proposed a simple ℓ_2 penalty, i.e. $R(\boldsymbol{x}) = \|\boldsymbol{x}\|^2 = \sum x_i^2$, which eventually led to the formal introduction of the *ridge* regression strategy by Hoerl and Kennard [34] 30 years later. Other important regularizers include the ℓ_0 penalty, $R(\boldsymbol{x}) = \|\boldsymbol{x}\|_0$, and the sparsity-promoting convex ℓ_1 relaxation $R(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, introduced by Shaobing and Donoho [46] in 1994 as *basis pursuit*, and by Tibshirani [49] in 1996 as the *least absolute shrinkage and selection operator* (LASSO). More generally, the ℓ_1 norm was introduced much earlier: as a penalty in 1969 [42], with specialized algorithms in 1973 [23], and as a robust loss in geophysics in 1973 [21]. In modern optimization, nonsmooth regularizers are widely used across a diverse set of applications, including in the training of neural network architectures [33]. Figure 1(a) illustrates the classic sparse regression iteration procedure for LASSO. Given the 1-norm of the solution, i.e. $\|\hat{\boldsymbol{x}}\|_1 = \tau$, the solution can be found by 'inflating' the level set of the data misfit until it intersects the ball $\mathbb{B}_1 \le \tau$. The geometry of the level sets influences both the robustness of the procedure with respect to noise, and the convergence rate of iterative algorithms used to find $\hat{\boldsymbol{x}}$.

Contributions: In this paper, we propose a broad framework for sparse relaxed regularized regression, called SR3. The key idea of SR3 is to solve a regularized problem that has three advantages over the state-of-the-art: (1) solutions are superior with respect to errors, false positives, and conditioning, (2) relaxation allows extremely fast algorithms for both convex and nonconvex formulations, and (3) the methods apply to composite regularizers. Rigorous theoretical results supporting these claims are presented in Section II. We demonstrate the advantages of SR3 (computational efficiency, higher accuracy, faster convergence rates, greater flexibility) across a range of regularized regression problems with synthetic and real data, including applications in compressed sensing, LASSO, matrix completion, TV regularization, and group sparsity using a range of test problems in Section III.

II. SR3 METHOD

Our goal is to improve the robustness, computational efficiency, and accuracy of sparse and nonsmooth formulations. We *relax* (1) using an auxiliary variable $\boldsymbol{w} \in \mathbb{R}^n$ that is forced to be close to $\mathbf{C}\boldsymbol{x}$. Relaxation was recently shown to be an efficient technique for dealing with the class of nonconvex-composite problems [57]. The general SR3 formulation modifies (1) to the following

$$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2 + \lambda R(\boldsymbol{w}) + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x}-\boldsymbol{w}\|^2, \tag{2}$$

where κ is a relaxation parameter that controls the gap between $\mathbf{C}\boldsymbol{x}$ and $\boldsymbol{w}$. Importantly, κ controls both the strength of the improvements to the geometry/regularity of the relaxed problem relative to the original and the fidelity of the relaxed problem to the original. To recover a relaxed version of LASSO, for example, we take $R(\cdot) = \|\cdot\|_1$ and $\mathbf{C} = \mathbf{I}$. The SR3 formulation allows non-convex ℓ_p "norms" with $p < 1$, as well as smoothly clipped absolute deviation (SCAD) [28], and easily handles linear composite regularizers. Two widely used examples that rely on compositions are compressed sensing formulations that use tight frames [25], and total variation (TV) regularization in image denoising [45].

In the convex setting, the formulation (2) fits into a class of problems studied by Bauschke *et al.* [5], who credit the natural alternating minimization algorithm to Acker and Prestel [1] in 1980, and the original alternating projections method to Cheney and Goldstein [20] in 1959 and Von Neumann in 1950 [53, Th. 13.7]. The main novelty of the SR3 approach is in using (2) to extract information from the $\boldsymbol{w}$ variable. We also allow nonconvex regularizers $R(\cdot)$, using the structure of (2) to simplify the analysis.

The success of SR3 stems from two key ideas. First, sparsity and accuracy requirements are split between $\boldsymbol{w}$ and $\boldsymbol{x}$ in the formulation (2), relieving the pressure these competing goals put on $\boldsymbol{x}$ in (1). Second, we can partially minimize (2) in $\boldsymbol{x}$ to obtain a function in $\boldsymbol{w}$ alone, with nearly spherical level sets, in contrast to the elongated elliptical level sets of $\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2$. In $\boldsymbol{w}$ coordinates, it is much easier to find the correct support. Figure 1(b) illustrates this advantage of SR3 on the LASSO problem.

A. SR3 AND VALUE FUNCTION OPTIMIZATION

Associated with (2) is a *value function* formulation that allows us to precisely characterize the relaxed framework. The value function is obtained by minimizing (2) in $\boldsymbol{x}$:

$$v(\boldsymbol{w}) := \min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2 + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x}-\boldsymbol{w}\|^2. \tag{3}$$

We assume that $\mathbf{H}_\kappa = \mathbf{A}^\top\mathbf{A} + \kappa\mathbf{C}^\top\mathbf{C}$ is invertible. Under this assumption, $\boldsymbol{x}(\boldsymbol{w}) = \mathbf{H}_\kappa^{-1}\left(\mathbf{A}^\top\boldsymbol{b} + \kappa\mathbf{C}^\top\boldsymbol{w}\right)$ is unique. We now define

$$\mathbf{F}_\kappa = \begin{bmatrix} \kappa\mathbf{A}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \\ \sqrt{\kappa}(\mathbf{I} - \kappa\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top) \end{bmatrix}, \quad \mathbf{F}_\kappa \in \mathbb{R}^{(m+n)\times n}$$

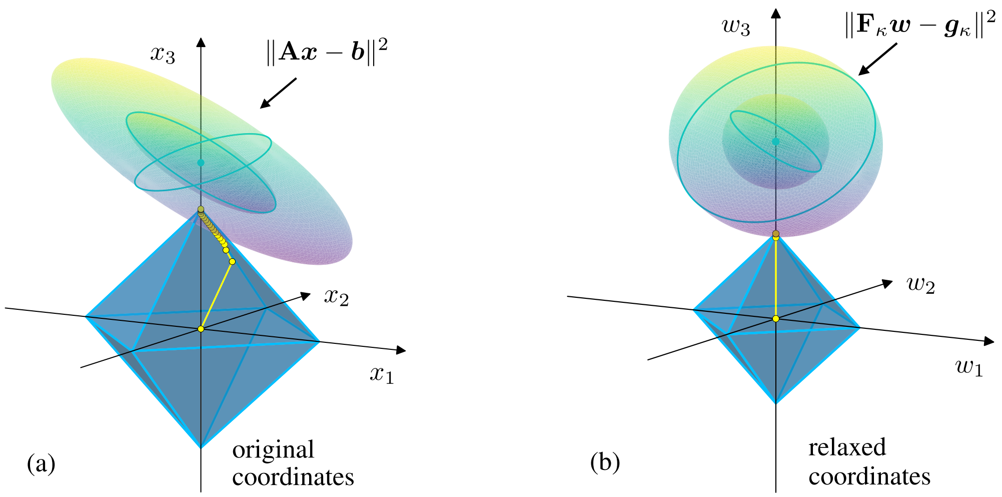


FIGURE 1. (a) Level sets (green ellipses) of the quadratic part of LASSO (1) and corresponding path of prox-gradient to the solution (40 iterations) in x-coordinates. (b) Level sets (green spheres) of the quadratic part of the SR3 value function (3) and corresponding SR3 solution path (2 iterations) in relaxed coordinates w. Blue octahedra show the ℓ_1 ball in each set of coordinates. SR3 decreases the singular values of $\mathbf{F}_\kappa$ relative to those of A with a weaker effect on the small ones, 'squashing' the level sets into approximate spheres, accelerating convergence, and improving performance.

$$\begin{aligned} \mathbf{G}_\kappa &= \begin{bmatrix} \mathbf{I}-\mathbf{A}\mathbf{H}_\kappa^{-1}\mathbf{A}^\top \\ \sqrt{\kappa}\,\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{A}^\top \end{bmatrix}, \quad \mathbf{G}_\kappa \in \mathbb{R}^{(m+n)\times m} \\ \boldsymbol{g}_\kappa &= \mathbf{G}_\kappa \boldsymbol{b}, \quad \boldsymbol{g}_\kappa \in \mathbb{R}^{m+n} \end{aligned} \tag{4}$$

which gives a closed form for (3):

$$v(\boldsymbol{w}) = \frac{1}{2}\|\mathbf{F}_\kappa \boldsymbol{w} - \boldsymbol{g}_\kappa\|^2.$$

Problem (2) then reduces to

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\mathbf{F}_\kappa \boldsymbol{w} - \boldsymbol{g}_\kappa\|^2 + \lambda R(\boldsymbol{w}). \tag{5}$$

The ellipsoid in Fig. 1(a) shows the level sets of $\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2$, while the spheroid in Fig. 1(b) shows the level sets of $\|\mathbf{F}_\kappa \boldsymbol{w} - \boldsymbol{g}_\kappa\|^2$. Partial minimization improves the conditioning of the problem, as seen in Figure 1, and can be characterized by a simple theorem.

Denote by $\sigma_i(\cdot)$ the function that returns the i-th largest singular value of the argument, with $\sigma_{\max}(\mathbf{A})$ denoting the largest singular value $\sigma_1(\mathbf{A})$, and $\sigma_{\min}(\mathbf{A})$ denoting the smallest (reduced) singular value $\sigma_{\min(m,d)}(\mathbf{A})$. Let $\operatorname{cond}(\mathbf{A}) := \sigma_{\max}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$ denote the condition number of $\mathbf{A}$. The following result relates singular values of $\mathbf{F}_\kappa$ to those of $\mathbf{A}$ and $\mathbf{C}$. Stronger results apply to the special cases $\mathbf{C} = \mathbf{I}$, which covers the Lasso, and $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, which covers compressed sensing formulations with tight frames ($\mathbf{C} = \mathbf{\Phi}^\top$ with $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$) [19], [25], [27].

Theorem 1: When $\lambda = 0$, (5) and (1) share the same solution set. We also have the following relations:

$$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa = \kappa\mathbf{I} - \kappa^2\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \tag{6}$$

$$\sigma_i(\mathbf{F}_\kappa^\top\mathbf{F}_\kappa) = \kappa - \kappa^2\sigma_{n-i+1}(\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top). \tag{7}$$

In addition, $\mathbf{0} \preceq \mathbf{F}_\kappa^\top\mathbf{F}_\kappa \preceq \kappa\mathbf{I}$ always, and when $n \geq d$ and $\mathbf{C}$ has full rank (i.e. $\mathbf{C}^\top\mathbf{C}$ is invertible), we have

$$\sigma_{\min}(\mathbf{F}_\kappa^\top\mathbf{F}_\kappa) \geq \frac{\sigma_{\min}(\mathbf{A}^\top\mathbf{A})/\sigma_{\max}(\mathbf{C}^\top\mathbf{C})}{1+\sigma_{\min}(\mathbf{A}^\top\mathbf{A})/(\kappa\sigma_{\max}(\mathbf{C}^\top\mathbf{C}))}.$$

When $\mathbf{C} = \mathbf{I}$, we have

$$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa = \mathbf{A}^\top(\mathbf{I}+\mathbf{A}\mathbf{A}^\top/\kappa)^{-1}\mathbf{A} \tag{8}$$

$$\sigma_i(\mathbf{F}_\kappa^\top\mathbf{F}_\kappa) = \frac{\sigma_i(\mathbf{A}^\top\mathbf{A})}{1+\sigma_i(\mathbf{A}^\top\mathbf{A})/\kappa}, \tag{9}$$

so that the condition numbers of $\mathbf{F}_\kappa$ and $\mathbf{A}$ are related by

$$\operatorname{cond}(\mathbf{F}_\kappa) = \operatorname{cond}(\mathbf{A})\sqrt{\frac{\kappa+\sigma_{\min}(\mathbf{A})^2}{\kappa+\sigma_{\max}(\mathbf{A})^2}}. \tag{10}$$

Theorem 1 lets us interpret (5) as a re-weighted version of the original problem (1). In the general case, the properties of $\mathbf{F}$ depend on the interplay between $\mathbf{A}$ and $\mathbf{C}$. The re-weighted linear map $\mathbf{F}_\kappa$ has superior properties to $\mathbf{A}$ in special cases. Theorem 1 gives strong results for $\mathbf{C} = \mathbf{I}$, and we can derive analogous results when $\mathbf{C}$ has orthogonal columns and full rank.

Corollary 1: Suppose that $\mathbf{C} \in \mathbb{R}^{n\times d}$ with $n \geq d$ and $\mathbf{C}^\top\mathbf{C} = \mathbf{I}_d$. Then,

$$\sigma_i(\mathbf{F}_\kappa) = \begin{cases} \sqrt{\kappa}\,\dfrac{\sigma_{i-(n-d)}(\mathbf{A})}{\sqrt{\kappa+\sigma_{i-(n-d)}(\mathbf{A})^2}} & i > n\text{-}d \\ \sqrt{\kappa} & i \leq n-d. \end{cases} \tag{11}$$

For $n > d$, this implies

$$\operatorname{cond}(\mathbf{F}_\kappa) = \operatorname{cond}(\mathbf{A})\sqrt{\frac{\kappa+\sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2}}. \tag{12}$$

Algorithm 1 SR3 for (2), Concept

```
**Input:** $w^0$
**Initialize:** $k = 0, \eta \leq \frac{1}{\kappa}$
**while** not converged **do**
    $k \leftarrow k + 1$
    $w^k \leftarrow \text{prox}_{\eta\lambda R}(w^{k-1} - \eta \mathbf{F}_\kappa^\top(\mathbf{F}_\kappa w^{k-1} - g_\kappa))$
**Output:** $w^k$
```

Algorithm 2 SR3 for (2), Implementation

```
**Input:** $w^0$
**Initialize:** $k = 0, \eta = \frac{1}{\kappa}$
**while** not converged **do**
    $k \leftarrow k + 1$
    $x^k \leftarrow \mathbf{H}_\kappa^{-1}\left(\mathbf{A}^\top b + \kappa \mathbf{C}^\top w^{k-1}\right)$
    $w^k \leftarrow \text{prox}_{\eta\lambda R}(\mathbf{C}x^k)$
**Output:** $w^k$
```

Algorithm 3 Prox-Gradient for (1)

```
**Input:** $x^0$
**Initialize:** $k = 0, \eta \leq \frac{1}{\sigma_{\max}(\mathbf{A})^2}$
**while** not converged **do**
    $k \leftarrow k + 1$
    $x^k \leftarrow \text{prox}_{\eta\lambda R(\mathbf{C}\cdot)}(x^{k-1} - \eta \mathbf{A}^\top(\mathbf{A}x^{k-1} - b))$
**Output:** $x^k$
```

When $n = d$, this implies

$$\text{cond}(\mathbf{F}_\kappa) = \text{cond}(\mathbf{A})\sqrt{\frac{\kappa + \sigma_{\min}(\mathbf{A})^2}{\kappa + \sigma_{\max}(\mathbf{A})^2}}. \tag{13}$$

Proof: Let $\bar{\mathbf{C}} = \begin{bmatrix}\mathbf{C} & \mathbf{C}^\perp\end{bmatrix}$ where the columns of $\mathbf{C}^\perp$ form an orthonormal basis for the orthogonal complement of the range of $\mathbf{C}$. Then, by Theorem 1,

$$\bar{\mathbf{C}}^\top \mathbf{F}_\kappa^\top \mathbf{F}_\kappa \bar{\mathbf{C}} = \begin{bmatrix}\mathbf{A}^\top(\mathbf{I} + \mathbf{A}\mathbf{A}^\top/\kappa)^{-1}\mathbf{A} & \\ & \kappa \mathbf{I}_{n-d}\end{bmatrix}. \tag{14}$$

The result follows from the second part of Theorem 1 . □

When $\mathbf{C}$ is a square orthogonal matrix, partial minimization of (3) shrinks the singular values of $\mathbf{F}_\kappa$ relative to $\mathbf{A}$, with less shrinkage for smaller singular values, which gives a smaller condition number as seen in Figure 1 for $\mathbf{C} = \mathbf{I}$. As a result, iterative methods for (5) converge much faster than the same methods applied to (1), especially for ill-conditioned $\mathbf{A}$. The geometry of the level sets of (5) also encourages the discovery of sparse solutions; see the path-to-solution for each formulation in Figure 1. The amount of improvement depends on the size of κ, with smaller values of κ giving better conditioned problems. For instance, consider setting $\kappa = (\sigma_{\max}(\mathbf{A})^2 - \sigma_{\min}(\mathbf{A})^2)/\mu^2$ for some $\mu > 1$. Then, by Corollary 1, $\text{cond}(\mathbf{F}_\kappa) \leq 1 + \text{cond}(\mathbf{A})/\mu$.

B. ALGORITHMS FOR THE SR3 PROBLEM

Problem (5) can be solved using a variety of algorithms, including the prox-gradient method detailed in Algorithm 1. In the convex case, Algorithm 1 is equivalent to the alternating method of [5]. The w update is given by

$$\hat{w}^{k+1} = \text{prox}_{\frac{\lambda}{\kappa}R}\left(w^k - \frac{1}{\kappa}\mathbf{F}_\kappa^\top(\mathbf{F}_\kappa w^k - g_\kappa)\right), \tag{15}$$

where $\text{prox}_{\frac{\lambda}{\kappa}R}$ is the *proximity operator* (prox) for R (see e.g. [22]) evaluated at $\mathbf{C}x$. The prox in Algorithm 1 is easy to evaluate for many important convex and nonconvex functions, often taking the form of a separable atomic operator, i.e. the prox requires a simple computation for each individual entry of the input vector. For example, $\text{prox}_{\lambda\|\cdot\|_1}$ is the *soft-thresholding* (ST) operator:

$$\text{prox}_{\lambda\|\cdot\|_1}(x)_i = \text{sign}(x_i)\max(|x_i| - \lambda, 0). \tag{16}$$

Algorithm 1 is a conceptual description of the proximal gradient algorithm applied to (5), and is analyzed below. However, **it is not necessary to form or apply $\mathbf{F}_\kappa$ to implement the algorithm**. To make this clear, a simpler, equivalent method that computes an explicit x^k in order to simplify the update for w^k is specified in Algorithm 2. The equivalence of these algorithms is shown in the Appendix.

In the pseudocode, partial minimization in x is written using the normal equations, but any least squares solution may be used. When $\mathbf{H}_\kappa$ cannot be directly inverted, (e.g. it is only available through its action on a vector or it is very large) Algorithm 2 admits an inexact modification, where an iterative method (e.g. conjugate gradient) is used to get x^k.

It is useful to contrast Algorithm 1 with the proximal gradient algorithm for the original problem (1), detailed in Algorithm 3. First, Algorithm 3 may be difficult to implement when $\mathbf{C} \neq \mathbf{I}$, as the prox operator may no longer be separable or atomic. An iterative algorithm is required to evaluate

$$\text{prox}_{\lambda\|\mathbf{C}\cdot\|_1}(x) = \arg\min_y \frac{1}{2\lambda}\|x - y\|^2 + \|\mathbf{C}y\|_1. \tag{17}$$

In contrast, Algorithm 1 always solves (5), which is regularized by $R(\cdot)$ rather than by a composition $R(\mathbf{C}\cdot)$, with $\mathbf{C}$ instead changing $\mathbf{F}_\kappa$ and g_κ, see (4). In the equivalent Algorithm 2, $\mathbf{C}$ affects the update of x^k, and w is updated using the prox of $R(\cdot)$ evaluated at the vector $\mathbf{C}x^k$. Viewing SR3 as a prox-gradient for the value function in Algorithm 1 has important consequences, since the prox-gradient method converges for a wide class of problems, including nonconvex regularizers [4]. For regularized least squares problems specifically, we derive a self-contained convergence theorem with a sublinear convergence rate.

Theorem 2 (Proximal Gradient Descent for Regularized Least Squares): Consider the linear regression objective,

$$\min_x \; p(x) := \frac{1}{2}\|\mathbf{A}x - b\|^2 + \lambda R(x) \,,$$

where p is bounded below, so that

$$-\infty < p^* = \inf_x p(x),$$

and R may be nonsmooth and nonconvex. With step $t = 1/\sigma_{\max}(\mathbf{A})^2$, the iterates generated by Algorithm 3 satisfy

$$\boldsymbol{v}_{k+1} := (\|\mathbf{A}\|_2^2\mathbf{I} - \mathbf{A}^\top\mathbf{A})(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) \in \partial p(\boldsymbol{x}_{k+1}),$$

i.e. $\boldsymbol{v}_{k+1}$ is an element of the subdifferential of $p(\boldsymbol{x})$ at the point $\boldsymbol{x}_{k+1}$,[1] *and*

$$\min_{k=0,\ldots N} \|\boldsymbol{v}_{k+1}\|^2 \le \frac{1}{N}\sum_{k=0}^{N-1}\|\boldsymbol{v}_{k+1}\|^2 \le \frac{\|\mathbf{A}\|_2^2}{N}(p(\boldsymbol{x}_0) - p^*)\,.$$

Therefore Algorithm 3 converges at a sublinear rate to a stationary point of p.

Theorem 2 always applies to the SR3 approach, which uses value function (5). When $\mathbf{C} = \mathbf{I}$, we can also compare the convergence rate of Algorithm 1 for (5) to the rate for Algorithm 3 for (3). In particular, the rates of Algorithm 1 are independent of $\mathbf{A}$ when $\mathbf{A}$ does not have full rank, and depend only weakly on $\mathbf{A}$ when $\mathbf{A}$ has full rank, as detailed in Theorem 3.

Theorem 3 Suppose that $\mathbf{C} = \mathbf{I}$. Let $\boldsymbol{x}^*$ and $\boldsymbol{w}^*$ denote the minimum values of $p_x(\boldsymbol{x}) := \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + R(\boldsymbol{x})$ and $p_w(\boldsymbol{w}) := \frac{1}{2}\|\mathbf{F}_\kappa\boldsymbol{w} - \boldsymbol{g}_\kappa\|^2 + R(\boldsymbol{w})$, respectively. Let $\boldsymbol{x}_k$ denote the iterates of Algorithm 3 applied to p_x, and $\boldsymbol{w}_k$ denote the iterates of Algorithm 1 applied to p_w, with step sizes $\eta_x = \frac{1}{\sigma_{\max}(\mathbf{A})^2}$ and $\eta_w = \frac{1}{\sigma_{\max}(\mathbf{F}_\kappa)^2}$. The iterates always satisfy

$$\begin{aligned} \boldsymbol{v}_{k+1}^x &= (\|\mathbf{A}\|_2^2\mathbf{I} - \mathbf{A}^\top\mathbf{A})(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) \in \partial p_x(\boldsymbol{x}_{k+1}) \\ \boldsymbol{v}_{k+1}^w &= (\kappa\mathbf{I} - \mathbf{F}^\top\mathbf{F})(\boldsymbol{w}_k - \boldsymbol{w}_{k+1}) \in \partial p_w(\boldsymbol{w}_{k+1}). \end{aligned}$$

For general R and any $\mathbf{A}$ we have the following rates:

$$\begin{aligned} \frac{1}{N}\sum_{k=0}^{N-1}\|\boldsymbol{v}_{k+1}^x\|^2 &\le \frac{\|\mathbf{A}\|_2^2}{N}(p_x(\boldsymbol{x}_0) - p_x^*) \\ \frac{1}{N}\sum_{k=0}^{N-1}\|\boldsymbol{v}_{k+1}^w\|^2 &\le \frac{\kappa}{N}(p_w(\boldsymbol{x}_0) - p_w^*). \end{aligned}$$

For convex R and any $\mathbf{A}$ we also have

$$\begin{aligned} \frac{p_x(\boldsymbol{x}) - p_x(\boldsymbol{x}^*)}{\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2} &\le \frac{\sigma_{\max}(\mathbf{A})^2}{2(k+1)} \\ \frac{p_w(\boldsymbol{w}) - p_w(\boldsymbol{w}^*)}{\|\boldsymbol{w}^0 - \boldsymbol{w}^*\|^2} &\le \frac{\sigma_{\max}(\mathbf{F}_\kappa)^2}{2(k+1)} \\ &\le \frac{\frac{\sigma_{\max}(\mathbf{A})^2}{1+\sigma_{\max}(\mathbf{A})^2/\kappa}}{2(k+1)} \le \frac{\kappa}{2(k+1)}. \end{aligned}$$

For convex R and $\mathbf{A}$ with full rank, we also have

$$\begin{aligned} \frac{\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2}{\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2} &\le \left(1 - \frac{\sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2}\right)^k \\ \frac{\|\boldsymbol{w}^k - \boldsymbol{w}^*\|^2}{\|\boldsymbol{w}^0 - \boldsymbol{w}^*\|^2} &\le \left(1 - \frac{\sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2}\frac{\sigma_{\max}(\mathbf{A})^2 + \kappa}{\sigma_{\min}(\mathbf{A})^2 + \kappa}\right)^k \end{aligned}$$

When $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, Algorithm 3 may not be implementable. However, SR3 is implementable, with rates equal to those for the $\mathbf{C} = \mathbf{I}$ case when $n = d$ and with rates as in the following corollary when $n > d$.

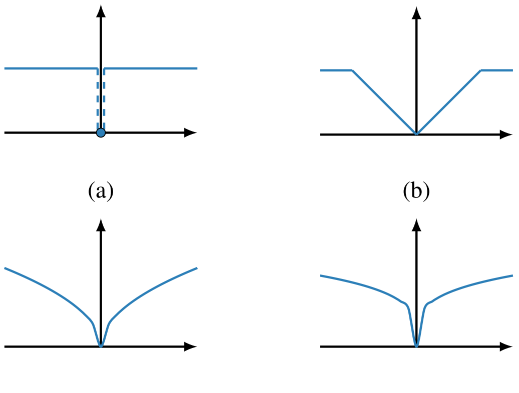


FIGURE 2. Nonconvex sparsity promoting regularizers. (a) ℓ_0 norm. (b) Clipped absolute deviation. (c) ℓ_p norm ($p = \frac{1}{2}$). (d) ℓ_p norm ($p = \frac{1}{4}$).

Corollary 2 When $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$ and $n > d$, let $\boldsymbol{w}^*$ denote the minimum value of $p_w(\boldsymbol{w}) := \frac{1}{2}\|\mathbf{F}_\kappa\boldsymbol{w} - \boldsymbol{g}_\kappa\|^2 + R(\boldsymbol{w})$, and let $\boldsymbol{w}_k$ denote the iterates of Algorithm 1 applied to p_w, with step size $\eta_w = \frac{1}{\kappa}$. The iterates always satisfy

$$\boldsymbol{v}_{k+1}^w = (\kappa\mathbf{I} - \mathbf{F}^\top\mathbf{F})(\boldsymbol{w}_k - \boldsymbol{w}_{k+1}) \in \partial p_w(\boldsymbol{w}_{k+1}).$$

For general R and any $\mathbf{A}$ we have the following rates:

$$\frac{1}{N}\sum_{k=0}^{N-1}\|\boldsymbol{v}_{k+1}^w\|^2 \le \frac{\kappa}{N}(p_w(\boldsymbol{x}_0) - p_w^*).$$

For convex R and any $\mathbf{A}$ we also have

$$\frac{p_w(\boldsymbol{w}) - p_w(\boldsymbol{w}^*)}{\|\boldsymbol{w}^0 - \boldsymbol{w}^*\|^2} \le \frac{\kappa}{2(k+1)}$$

For convex R and $\mathbf{A}$ with full rank, we also have

$$\frac{\|\boldsymbol{w}^k - \boldsymbol{w}^*\|^2}{\|\boldsymbol{w}^0 - \boldsymbol{w}^*\|^2} \le \left(1 - \frac{\sigma_{\min}(\mathbf{A}^\top\mathbf{A})}{\kappa + \sigma_{\min}(\mathbf{A}^\top\mathbf{A})}\right)^k$$

Algorithm 1 can be used with both convex and nonconvex regularizers, as long as the prox operator of the regularizer is available. A growing list of proximal operators is reviewed by [22]. Notable nonconvex prox operators in the literature include (1) indicator of set of rank r matrices, (2) spectral functions (with proximable outer functions) [26], [38], (3) indicators of unions of convex sets (project onto each and then choose the closest point), (4) MCP penalty [56], (5) firm-thresholding penalty [29], and (6) indicator functions of finite sets (e.g., $x \in \{-1, 0, 1\}^d$). Several nonconvex prox operators specifically used in sparse regression are detailed in the next section.

C. NONCONVEX REGULARIZERS AND CONSTRAINTS

1) NONCONVEX REGULARIZERS: ℓ_0

The 1-norm is often used as a convex alternative to ℓ_0, defined by $\|\boldsymbol{x}\|_0 = |\{i : x_i \neq 0\}|$, see panel (a) of Figure 2. The nonconvex ℓ_0 has a simple prox — hard thresholding (HT) [9],

[1] For nonconvex problems, the subdifferential must be carefully defined; see the preliminaries in the Appendix.

TABLE 1. Proximal operators of sparsity-promoting regularizers.

$R(\boldsymbol{x})$	$r(x)$	$\text{prox}_{\alpha r}(z)$	Solution
$\|\boldsymbol{x}\|_1$	$\lvert x\rvert$	$\begin{cases}\text{sign}(z)(\lvert z\rvert-\alpha), & \lvert z\rvert>\alpha\\ 0, & \lvert z\rvert\le\alpha\end{cases}$	Analytic
$\|\boldsymbol{x}\|_0$	$\begin{cases}1, & x\neq 0\\ 0, & x=0\end{cases}$	$\begin{cases}0, & \lvert z\rvert\le\sqrt{2\alpha}\\ z, & \lvert z\rvert>\sqrt{2\alpha}\end{cases}$	Analytic
$\|\boldsymbol{x}\|_p^p$ $(p<1)$	$\lvert x\rvert^p$	see Appendix	Coordinate-wise Newton
$\text{CAD}(\boldsymbol{x};\rho)$	$\begin{cases}\lvert x\rvert, & \lvert x\rvert\le\rho\\ \rho, & \lvert x\rvert>\rho\end{cases}$	$\begin{cases}z, & \lvert z\rvert>\rho\\ \text{sign}(z)(\lvert z\rvert-\alpha), & \alpha<\lvert z\rvert\le\rho\\ 0, & \lvert z\rvert\le\alpha\end{cases}$	Analytic

see Table 1. The SR3 formulation with the ℓ_0 regularizer uses HT instead of the ST operator (16) in line 5 of Algorithm 1.

2) NONCONVEX REGULARIZERS: ℓ_p^p FOR $P \in (0, 1)$

The ℓ_p^p regularizer for $p \in (0, 1)$ is often used for sparsity promotion, see e.g. [36] and the references within. Two members of this family are shown in panels (c) and (d) of Figure 2. The ℓ_p^p prox subproblem is given by

$$\min_x f_{\alpha,p}(x; z) := \frac{1}{2\alpha}(x-z)^2 + |x|^p \tag{18}$$

This problem is studied in detail by [18]. Closed form solutions are available for special cases $p \in \left\{\frac{1}{2}, \frac{2}{3}\right\}$; but a provably convergent Newton method is available for all p. Using a simple method *for each coordinate*, we can globally solve the nonconvex problem (18) [18, Proposition 8]. Our implementation is summarized in the Appendix. The $\ell_{1/2}$ regularizer is particularly useful for CS, and is known to do better than either ℓ_0 or ℓ_1.

3) NONCONVEX REGULARIZERS: (S)CAD

The (Smoothly) Clipped Absolute Deviation (SCAD) [28] is a sparsity promoting regularizer used to reduce bias in the computed solutions. A simple un-smoothed version (CAD) appears in panel (b) of Figure 2, and the analytic prox is given in Table 1. This regularizer, when combined with SR3, obtains the best results in the CS experiments in Section III.

4) COMPOSITE REGULARIZATION: TOTAL VARIATION (TV)

TV regularization can be written as $\text{TV}(\boldsymbol{x}) = R(\mathbf{C}\boldsymbol{x}) = \|\mathbf{C}\boldsymbol{x}\|_1$, with $\mathbf{C}$ a (sparse) difference matrix (see (23)). The SR3 formulation is solved by Algorithm 1, a prox-gradient (primal) method. In contrast, most TV algorithms use primal-dual methods because of the composition $\|\mathbf{C}\boldsymbol{x}\|_1$ [16].

5) CONSTRAINTS AS INFINITE-VALUED REGULARIZERS

The term $R(\cdot)$ does not need to be finite valued. In particular, for any set C that has a projection, we can take $R(\cdot)$ to be the indicator function of C, given by

$$R_C(\boldsymbol{x}) = \begin{cases} 0 & \boldsymbol{x} \in C \\ \infty & \boldsymbol{x} \notin C. \end{cases},$$

so that $\text{prox}_R(\boldsymbol{x}) = \text{proj}_C(\boldsymbol{x})$. Simple examples of such regularizers include convex non-negativity constraints ($\boldsymbol{x} \geq 0$) and nonconvex spherical constraints ($\|\boldsymbol{x}\|_2 = r$).

D. OPTIMALITY OF SR3 SOLUTIONS

We now consider the relationship between the optimal solution $\hat{\boldsymbol{w}}$ to problem (5), and the original problem (1).

Theorem 4 (Optimal Ratio): Assume $\mathbf{C} = \mathbf{I}$, *and let* λ_1 *for* (1) *and* λ_2 *for* (5) *be related by the ratio* $\tau = \lambda_2/\lambda_1$, *and let* $\hat{\boldsymbol{w}}^k$ *be the optimal solution for* (5) *with parameter* λ_2. *If* λ_2 *is set to be* $\tau\lambda_1$ *where*

$$\hat{\tau} = \mathop{arg\,min}_{\tau>0} \; \|\tau\mathbf{I} - \kappa\mathbf{H}_\kappa^{-1}\|_2 = \frac{\kappa}{2}(\sigma_{\max}(\mathbf{H}_\kappa^{-1}) + \sigma_{\min}(\mathbf{H}_\kappa^{-1})),$$

then the distance to optimality of $\hat{\boldsymbol{w}}^1$ *for* (1) *is no more than*

$$\frac{\sigma_{\max}(\mathbf{A})^2 - \sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2 + \sigma_{\min}(\mathbf{A})^2 + 2\kappa}\|\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}\|.$$

Theorem 4 gives a way to choose λ_2 given λ_1 so that $\hat{\boldsymbol{w}}$ is as close as possible to the stationary point of (1), and characterizes the distance of $\hat{\boldsymbol{w}}$ to optimality of the original problem. The proof is given in the Appendix.

Theorem 4 shows that as κ increases, the solution $\hat{\boldsymbol{w}}$ moves closer to being optimal for the original problem (1). On the other hand, Theorem 3 suggests that lower κ values regularize the problem, making it easier to solve. In practice, we find that $\hat{\boldsymbol{w}}$ is useful and informative in a range of applications with moderate values of κ, see Section III.

III. RESULTS

The formulation (1) covers many standard problems, including variable selection (LASSO), compressed sensing, TV-based image de-noising, and matrix completion, shown in Fig. 3. In this section, we demonstrate the general flexibility of the SR3 formulation and its advantages over other state-of-the-art techniques. In particular, SR3 is faster than competing algorithms, and $\boldsymbol{w}$ is far more useful in identifying the support of sparse signals, particularly when data are noisy and $\mathbf{A}$ is ill-conditioned.

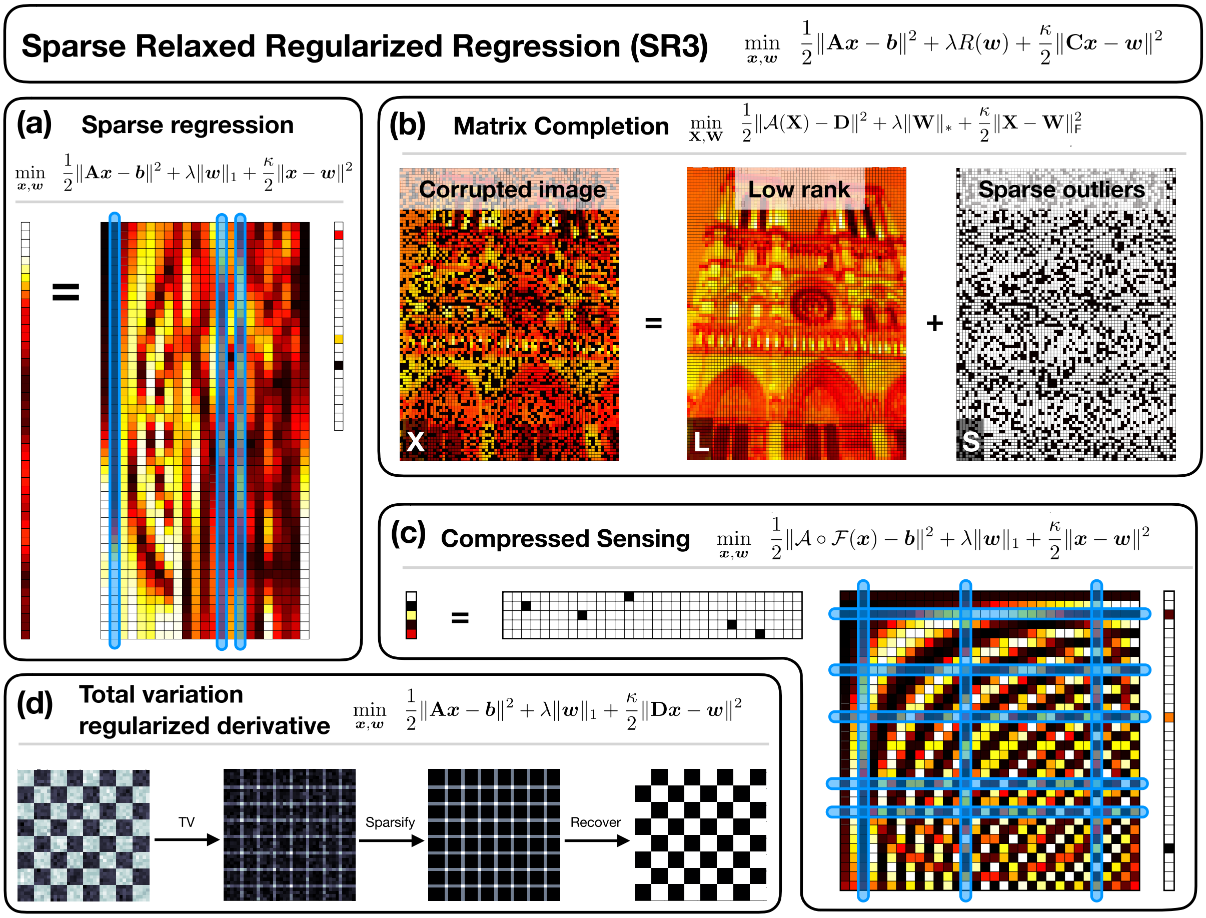

Sparse Relaxed Regularized Regression (SR3) $\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda R(\boldsymbol{w})+\frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x}-\boldsymbol{w}\|^2$
(a) Sparse regression
$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{w}\|_1+\frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{w}\|^2$
=
(b) Matrix Completion $\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathcal{A}(\mathbf{X})-\mathbf{D}\|^2+\lambda\|\mathbf{W}\|_*+\frac{\kappa}{2}\|\mathbf{X}-\mathbf{W}\|_F^2$
Corrupted image
Low rank
Sparse outliers
X
=
L
+
S
(c) Compressed Sensing $\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathcal{A}\circ\mathcal{F}(\boldsymbol{x})-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{w}\|_1+\frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{w}\|^2$
=
(d) Total variation regularized derivative $\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{w}\|_1+\frac{\kappa}{2}\|\mathbf{D}\boldsymbol{x}-\boldsymbol{w}\|^2$
TV
Sparsify
Recover

FIGURE 3. Common optimization applications where the SR3 method improves performance. For each method, the specific implementation of our general architecture (2) is given.

A. SR3 VS. LASSO AND COMPRESSED SENSING

Using Eqs. (1) and (2), the LASSO and associated SR3 problems are

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{x}\|_1 \tag{19}$$

$$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{w}\|_1+\frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{w}\|^2 \tag{20}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$. LASSO is often used for variable selection, i.e. finding a sparse set of coefficients $\boldsymbol{x}$ that correspond to variables (columns of $\mathbf{A}$) most useful for predicting the observation $\boldsymbol{b}$. We compare the quality and numerical efficiency of Eqs. (19) and (20). The formulation in (20) is related to an earlier sequentially thresholded least squares algorithm that was used for variable selection to identify nonlinear dynamical systems from data [11].

In all LASSO experiments, observations are generated by $\boldsymbol{b} = \mathbf{A}\boldsymbol{x}_t + \sigma\boldsymbol{\epsilon}$, where $\boldsymbol{x}_t$ is the true signal, and $\boldsymbol{\epsilon}$ is independent Gaussian noise.

1) LASSO PATH

The LASSO path refers to the set of solutions obtained by sweeping over λ in (1) from a maximum λ, which gives $\boldsymbol{x} = \mathbf{0}$, down to $\lambda = 0$, which gives the least squares solution. In [48], it was shown that (19) makes mistakes early along this path.

Problem Setup: As in [48], the measurement matrix $\mathbf{A}$ is 1010×1000, with entries drawn from $\mathcal{N}(0, 1)$. The first 200 elements of the true solution $\boldsymbol{x}_t$ are set to be 4 and the rest to be 0; $\sigma = 1$ is used to generate $\boldsymbol{b}$. Performing a λ sweep, we track the fraction of incorrect nonzero elements in the last 800 entries vs. the fraction of nonzero elements in the first 200 entries of each solution, i.e. the false discovery proportion (FDP) and true positive proportion (TPP).

Parameter Selection: We fix $\kappa = 100$ for SR3. Results are presented across a λ-sweep for both SR3 and LASSO.

Results: The results are shown in the top-right panel of Fig. 4. LASSO makes mistakes early along the path [48]. In contrast, SR3 recovers the support without introducing any false positives along the entire path until overfitting sets in with the 201st nonzero entry.

2) ROBUSTNESS TO NOISE

Observation noise makes signal recovery more difficult. We conduct a series of experiments to compare the robustness with respect to noise of SR3 with LASSO.

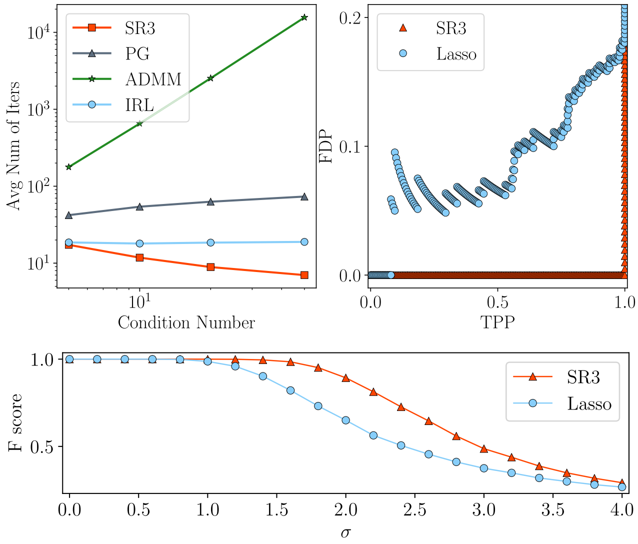


FIGURE 4. **Top Left: SR3 approach (red) is orders of magnitude faster than ADMM (green) or other first-order methods such as prox-gradient (gray). While IRL (blue) requires a comparable number of iterations, its cost per iteration is more expensive than SR3. Top Right: True Positives vs. False Positives along the LASSO path (blue) and along the SR3 path (red). Bottom: F_1 score of SR3 (red) and LASSO formulation (blue) with respect to different noise levels.**

Problem Setup: We choose our sensing matrix with dimension 200 by 500 and elements drawn independently from a standard Gaussian distribution. The true sparse signal has 20 non-zero entries, and we consider a range of noise levels $\sigma \in \{0.2i : i = 0, 1, \ldots, 20\}$. For each σ, we solve (19) and (20) for 200 different random trials. We record the F_1-score, $F_1 = 2(\text{precision} \cdot \text{recall})/(\text{precision} + \text{recall})$, to compare reconstruction quality. In the experiments, any entry in $\boldsymbol{x}$ which is greater than 0.01 is considered non-zero for the purpose of defining the recovered support.

Parameter Selection: We fix $\kappa = 100$ and perform a λ-sweep for both (19) and (20) to record the best F_1-score achievable by each method.

Results: We plot the average normalized F_1-score for different noise levels in the bottom panel of Fig. 4. SR3 has a uniformly higher F_1-score across all noise levels.

3) COMPUTATIONAL EFFICIENCY

We compare the computational efficiency of the Alternating Directions Method of Multipliers (ADMM) (see e.g. [10], [32]), proximal gradient algorithms (see e.g. [22]) on (19) with Algorithm 1, and a state-of-the-art Iteratively Reweighted Least-Squares (IRL) method, specifically IRucLq-v as in [36].

Problem Setup: We generate the observations with $\sigma = 0.1$. The dimension of $\mathbf{A}$ is 600 × 500, and we vary the condition number of the matrix $\mathbf{A}$ from 1 to 100. For each condition number, we solve the problem 10 times and record the average number of iterations required to reach a specified tolerance. We use the distance between the current and previous iteration to detect convergence for all algorithms. When the measure is less than a tolerance of 10^{-5} we terminate the algorithms.

TABLE 2. **Complexity Comparison for $A \in \mathbb{R}^{m\times n}$, $m \geq n$.**

Method	One-time Overhead	Cost of generic iteration
PG	—	$O(mn)$
ADMM	$O(mn^2 + n^3)$	$O(n^2)$
IRucLq-v	—	$O(mn^2 + n^3)$
SR3	$O(mn^2 + n^3)$	$O(n^2)$

Parameter Selection: We choose $\kappa = 1$, λ in (19) to be $\|\mathbf{A}^\top \boldsymbol{b}\|_\infty/5$, and λ in (20) to be $\|\mathbf{F}_\kappa^\top \boldsymbol{g}_\kappa\|_\infty/5$.

Results: The results (by number of iterations) are shown in the top left panel of Fig. 4. The complexity of each iteration is given in Table 2. The generic iterations of PG, ADMM, and SR3 have nearly identical complexity, with ADMM and SR3 requiring a one-time formation and factorization of an $n \times n$ matrix. IRucLq-v requires the formation and inversion of such a matrix at each iteration. From Fig. 4, SR3 requires far fewer iterations than ADMM and the proximal gradient method, especially as cond($\mathbf{A}$) increases. SR3 and IRucLq-v require a comparable number of iterations. A key difference is that ADMM requires dual variables, while SR3 is fundamentally a primal-only method. When cond($\mathbf{A}$) = 50, ADMM needs almost 10^4 iterations to solve (19); proximal gradient descent requires 10^2 iterations; and SR3 needs 10 iterations to solve (20). Overall, the SR3 method takes by far the least total compute time as the condition number increases. Further experiments, particularly for larger systems where iterative methods are needed, are left to future work.

4) SR3 FOR COMPRESSED SENSING

When $m \ll n$, the variable selection problem targeted by (19) is often called *compressed sensing* (CS). Sparsity is required to make the problem well-posed, as (19) has infinitely many solutions with $\lambda = 0$. In CS, columns of $\mathbf{A}$ are basis functions, e.g. the Fourier modes $A_{ij} = \exp(\boldsymbol{i}\alpha_j t_i)$, and $\boldsymbol{b}$ may be corrupted by noise [13]. In this case, compression occurs when m is smaller than the number of samples required by the Shannon sampling theorem.

Finding the optimal sparse solution is inherently combinatorial, and brute force solutions are only feasible for small-scale problems. In recent years, a series of powerful theoretical tools have been developed in [13]–[15], [24], and [25] to analyze and understand the behavior of (1) with $R(\cdot) = \|\cdot\|_1$ as a sparsity-promoting penalty. The main theme of these works is that if there is sufficient incoherence between the measurements and the basis, then exact recovery is possible. One weakness of the approach is that the incoherence requirement — for instance, having a small restricted isometry constant (RIC) [15] — may not be satisfied by the given samples, leading to sub-optimal recovery.

Problem Setup: We consider two synthetic CS problems. The sparse signal has dimension $d = 500$ and $k = 20$ nonzero coefficients with uniformly distributed positions and values randomly chosen as -2 or 2. In the first experiment,

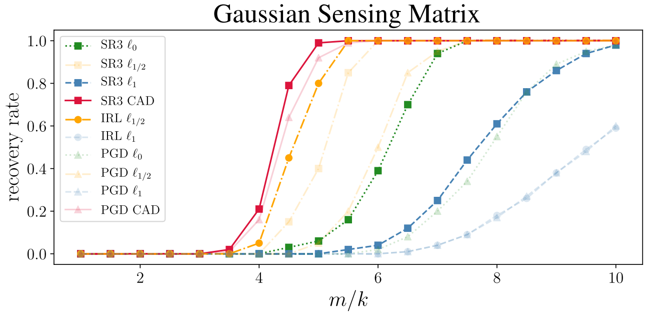


FIGURE 5. **Compressed sensing results: recovering a 20-sparse signal in $\mathbb{R}^{500}$ from a small number of measurements. We plot the recovery rate as the number of measurements increases. Line color and style are determined by the regularizer while marker shapes are determined by the algorithm/formulation used. For readability, only the best performing algorithm for each regularizer is plotted in bold, with the rest opaque. Left panel: the sensing matrix A has Gaussian entries. Nonconvex regularizers are in general more effective than convex regularizers. SR3 is the most effective formulation for each regularizer aside from $\ell_{1/2}$ for which the standard formulation with the IRucLq-v algorithm is best. SR3 CAD achieves a better final result compared to $\ell_{1/2}$ with IRLucLq-v. Right panel: the sensing matrix A has uniform entries. The traditional convex approaches fail dramatically as there is no longer a RIP-like condition. Even for the nonconvex regularizers, IRucLq-v shows significant performance degradation, while proximal gradient descent never succeeds. However, SR3 approaches still succeed, with only a minor efficiency gap (with respect to m/k) compared to the easier conditions in the left panel.**

the entries of $\mathbf{A} \in \mathbb{R}^{m\times n}$ are drawn independently from a normal distribution, which will generally have a small RIC [15] for sufficiently large m. In the second experiment, entries of $\mathbf{A} \in \mathbb{R}^{m\times n}$ are drawn from a uniform distribution on the interval $[0, 1]$, which are generally more coherent than using Gaussian entries.

In the classic CS context, recovering the support of the signal (indices of non-zero coefficients) is the main goal, as the optimal coefficients can be computed in a post-processing step. In the experiments, any entry in $\boldsymbol{x}$ which is greater than 0.01 is considered non-zero for the purpose of defining the recovered support. To test the effect of the number of samples m on recovery, we take measurements with additive Gaussian noise of the form $\mathcal{N}(0, 0.1)$, and choose m ranging from k to $20k$. For each choice of m we solve (1) and (2) 200 times. We compare results from 10 different formulations and algorithms: sparse regression with ℓ_0, $\ell_{1/2}$, ℓ_1 and CAD regularizers using PG; SR3 reformulations of these four problems using Algorithm 1, and sparse regression with $\ell_{1/2}$ and ℓ_1 regularizers using IRucLq-v.

Parameter Selection: For each instance, we perform a grid search on λ to identify the correct non-zero support, if possible. The fraction of runs for which there is a λ with successful support recovery is recorded. For all experiments we fix $\kappa = 5$, and we set $\rho = 0.5$ for the CAD regularizer.

Results: As shown in Figure 5, for relatively incoherent random Gaussian measurements, both the standard formulation (1) and SR3 succeed, particularly with the nonconvex regularizers. CAD$(\cdot, \rho)$, which incorporates some knowledge of the noise level in the parameter ρ, performs the best as a regularizer, followed by $\ell_{1/2}$, ℓ_0, and ℓ_1. The SR3 formulation obtains a better recovery rate for each m for most regularizers, with the notable exception of $\ell_{1/2}$. The IRucLq-v algorithm (which incorporates some knowledge of the sparsity level as an internal parameter) is the most effective method for $\ell_{1/2}$ regularization for such matrices.

For more coherent uniform measurements, SR3 obtains a recovery rate which is only slightly degraded from that of the Gaussian problem, while the results using (1) degrade drastically. In this case, SR3 is the most effective approach for each regularizer and provides the only methods which have perfect recovery at a sparsity level of $m/k \leq 10$, namely SR3-CAD, SR3-$\ell_{1/2}$, and SR3-ℓ_0.

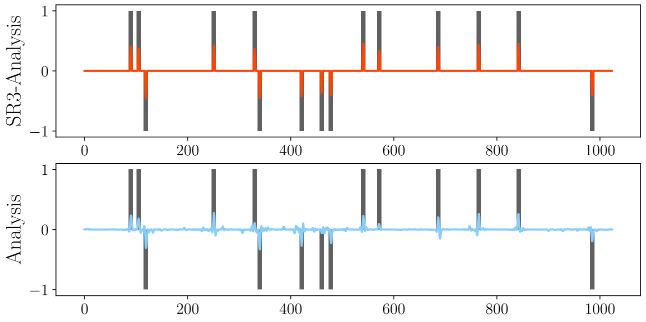


FIGURE 6. **Comparison of standard analysis with SR3-analysis. Top panel: result using SR3-analysis, plotting the final w (red) against the true signal (dark grey). Bottom panel: result using standard analysis and the IRL-D algorithm, plotting final $\mathbf{C}x$ (blue) against the true signal (dark grey).**

Remark: Many algorithms focus on the noiseless setting in compressive sensing, where the emphasis shifts to recovering signals that may have very small amplitudes [36]. SR3 is not well suited to this setting, since the underlying assumption is that $\boldsymbol{w}$ is near to $\boldsymbol{x}$ in the least squares sense.

5) ANALYSIS VS. SYNTHESIS

Compressive sensing formulations fall into two broad categories, analysis (21) and synthesis (22) (see [19], [27]):

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + R(\mathbf{C}\boldsymbol{x}), \tag{21}$$

$$\min_{\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{A}\mathbf{C}^\top\boldsymbol{\xi} - \boldsymbol{b}\|^2 + R(\boldsymbol{\xi}), \tag{22}$$

where $\mathbf{C}$ is the *analyzing operator*, $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\xi} \in \mathbb{R}^n$, and we assume $n \gg d$. In this section, we consider $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, i.e. $\mathbf{C}^\top$ is a tight frame. Synthesis represents $\boldsymbol{x}$ using the over-determined system $\mathbf{C}^\top$, and recovers the coefficients

$\boldsymbol{\xi}$ using sparse regression. Analysis directly works over the domain of the underlying signal $\boldsymbol{x}$ with the prior that $\mathbf{C}\boldsymbol{x}$ is sparse. The two methods are equivalent when $n \le d$, and very different when $n > d$ [19]. Both forms appear in a variety of inverse problems including denoising, interpolation and super-resolution. The work of [27] presents a thorough comparison of (21) and (22) across a range of signals, and finds that the effectiveness of each depends on problem type.

The SR3 formulation can easily solve both analysis and synthesis formulations. We have focused on synthesis thus far, so in this section we briefly consider analysis (21), under the assumption that $\mathbf{C}\boldsymbol{x}$ is almost sparse. When $l \gg d$, the analysis problem is formulated over a lower dimensional space. However, since $\mathbf{C}\boldsymbol{x}$ is always in the range of $\mathbf{C}$, it can never be truly sparse. If a sparse set of coefficients is needed, analysis formulations use post-processing steps such as thresholding. SR3, in contrast, can extract the sparse transform coefficients directly from the w variable. We compare SR3 with the Iteratively Reweighted Least-Squares-type algorithm IRL-D proposed by [35] for solving (21).

Problem Setup: We choose our dimensions to be $n = 1024$, $d = 512$ and $m = 128$. We generate the sensing matrix $\mathbf{A}$ with independent Gaussian entries and the true sparse coefficient $\boldsymbol{\xi}_t$ with 15 non-zero elements randomly selected from the set $\{-1, 1\}$. The true underlying signal is $\boldsymbol{x}_t = \mathbf{C}^\top \boldsymbol{\xi}$ and the measurements are generated by $\boldsymbol{b} = \mathbf{A}\boldsymbol{x}_t + \sigma\boldsymbol{\epsilon}$, where $\sigma = 0.1$ and $\boldsymbol{\epsilon}$ has independent Gaussian entries. We use ℓ_1 as the regularizer, $R(\cdot) = \lambda\|\cdot\|_1$.

Parameter Selection: We set κ for SR3 to be 5, λ for SR3 to be $\|\mathbf{F}_\kappa^\top \boldsymbol{g}_\kappa\|_\infty/2$, and $\|\mathbf{A}^\top \boldsymbol{b}\|_\infty/10$ for IRL-D. The λs are chosen to achieve the clearest separation between active and inactive signal coefficients for each method.

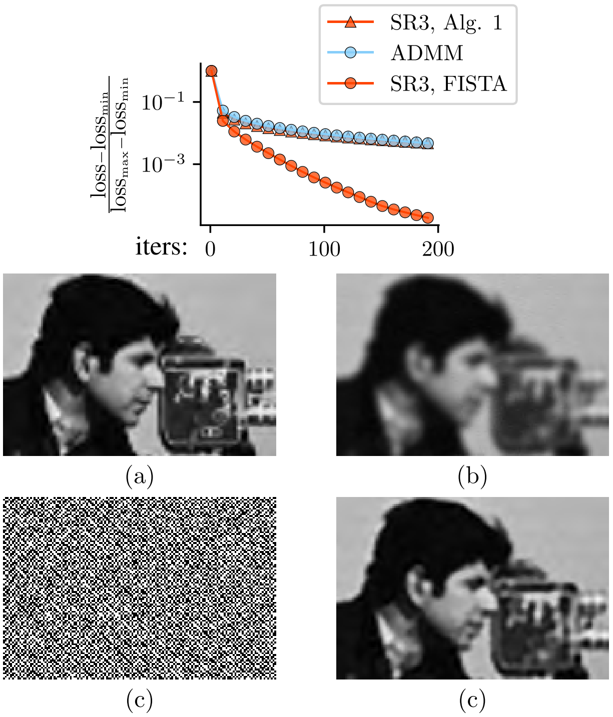
Results: The results are shown in Figure 6. The $\boldsymbol{w}$ in the SR3 analysis formulation is able to capture the support of the true signal cleanly, while $\mathbf{C}\boldsymbol{x}$ from the (21) identifies the support but is not completely sparse, requiring post-processing steps such as thresholding to get a support estimate.

B. SR3 FOR TOTAL VARIATION REGULARIZATION

Natural images are effectively modeled as large, smooth features separated by a few sparse edges. It is common to regularize ill-posed inverse problems in imaging by adding the so-called total variation (TV) regularization [7], [16], [17], [40], [45], [47], [54]. Let X_{ij} denote the i,j pixel of an $m \times n$ image. For convenience, we treat the indices as doubly periodic, i.e. $X_{i+pm,j+qn} = X_{i,j}$ for $p, q \in \mathbb{Z}$. Discrete x and y derivatives are defined by $[\mathbf{D}_x\mathbf{X}]_{ij} = X_{i+1,j} - X_{ij}$ and $[\mathbf{D}_y\mathbf{X}]_{ij} = X_{i,j+1} - X_{ij}$, respectively. The (isotropic) total variation of the image is then given by the sum of the length of the discrete gradient at each pixel, i.e.

$$R_{\mathrm{TV}}\begin{pmatrix}\mathbf{D}_x\mathbf{X}\\ \mathbf{D}_y\mathbf{X}\end{pmatrix} := \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{[\mathbf{D}_x\mathbf{X}]_{ij}^2 + [\mathbf{D}_y\mathbf{X}]_{ij}^2}. \tag{23}$$

Adding the TV regularizer (23) to a regression problem corresponds to imposing a sparsity prior on the discrete gradient.



(a)

(b)

(c)

(c)

FIGURE 7. **The top plot compares the progress of the SR3 and ADMM-type algorithms in reducing their losses, showing similar rates of convergence; accelerating SR3 with FISTA using Algorithm 4 yields an immediate significant improvement. Panels (a) and (b) show a detail of the original cameraman image and the image corrupted as described in the text, respectively. The incredibly noisy image resulting from inverting the blur without regularization ($\lambda = 0$) is shown in panel (c) and the crisper image resulting from the regularized SR3 problem (with $\lambda = .075$) is shown in panel (d) (the image resulting from the ADMM type algorithm of [16] is visually similar, with a similar SNR).**

Consider image deblurring (Fig. 7). The two-dimensional convolution $\mathbf{Y} = \mathbf{A} * \mathbf{X}$ is given by the sum $Y_{ij} = \sum_{p=1}^{m}\sum_{q=1}^{n} A_{pq}X_{i-p,j-q}$. Such convolutions are often used to model photographic effects, like distortion or motion blur. Even when the kernel $\mathbf{A}$ is known, the problem of recovering $\mathbf{X}$ given the blurred measurement is unstable because measurement noise is sharpened by 'inverting' the blur. Suppose that $\mathbf{B} = \mathbf{A}*\mathbf{X}+\nu\mathbf{G}$, where $\mathbf{G}$ is a matrix with entries given by independent entries from a standard normal distribution and ν is the noise level. To regularize the problem of recovering $\mathbf{X}$ from the corrupted signal $\mathbf{B}$, we add the TV regularization:

$$\hat{\mathbf{X}} = \mathop{arg\,min}_{\mathbf{X}} \frac{1}{2}\|\mathbf{A} * \mathbf{X} - \mathbf{B}\|_F^2 + \lambda R_{\mathrm{TV}}\begin{pmatrix}\mathbf{D}_x\mathbf{X}\\ \mathbf{D}_y\mathbf{X}\end{pmatrix}. \tag{24}$$

The natural SR3 reformulation is given by

$$\min_{\mathbf{X},\boldsymbol{w}_x,\boldsymbol{w}_y} \frac{1}{2}\|\mathbf{A} * \mathbf{X} - \mathbf{B}\|_F^2 + \lambda R_{\mathrm{TV}}\begin{pmatrix}\boldsymbol{w}_x\\ \boldsymbol{w}_y\end{pmatrix} + \frac{\kappa}{2}\left\|\begin{matrix}\boldsymbol{w}_x - \mathbf{D}_x\mathbf{X}\\ \boldsymbol{w}_y - \mathbf{D}_y\mathbf{X}\end{matrix}\right\|_F^2. \tag{25}$$

Problem Setup: In this experiment, we use the standard Gaussian blur kernel of size k and standard deviation σ, given by $A_{ij} = \exp\left(-(i^2 + j^2)/(2\sigma^2)\right)$, when $|i| < k$ and $|j| < k$, with the rest of the entries of $\mathbf{A}$ determined by periodicity or equal to zero. The signal $\mathbf{X}$ is the classic "cameraman" image of size 512×512. As a measure of the progress of a given method toward the solution, we evaluate the current loss at each iteration (the value of either the right hand side of (24) or (25)).

Parameter Selection: We set $\sigma = 2$, $k = 4$, $\nu = 2$, and $\lambda = 0.075$. The value of λ was chosen by hand to achieve reasonable image recovery. For SR3, we set $\kappa = 0.25$.

Results: Figure 7 demonstrates the stabilizing effect of TV regularization. Panels (a) and (b) show a detail of the image, i.e. $\mathbf{X}$, and the corrupted image, i.e. $\mathbf{B}$, respectively. In panel (c), we see that simply inverting the effect of the blur results in a meaningless image. Adding TV regularization gives a more reasonable result in panel (d).

Algorithm 4 FISTA for SR3 TV

```
Input: $w^0$
Initialize: $k = 0, a_0 = 1, v_0 = w^0, \eta \leq \frac{1}{\kappa}$
while not converged do
    $k \leftarrow k + 1$
    $v_k \leftarrow \mathrm{prox}_{\eta R}(w^{k-1} - \eta(\mathbf{F}_\kappa^\top(\mathbf{F}_\kappa w^{k-1} - g_\kappa)))$
    $a_k \leftarrow (1 + \sqrt{1 + 4a_{k-1}^2})/2$
    $w^k \leftarrow v_k + (a_{k-1} - 1)/a_k(v_k - v_{k-1})$
Output: $w^k$
```

In the top plot of Fig. 7, we compare SR3 and a primal-dual algorithm [16] on the objectives (25) and (24), respectively. Algorithm 1 converges as fast as the state-of-the-art method of [16]; it is not significantly faster because for TV regularization, the equivalent of the map $\mathbf{C}$ does not have orthogonal columns (so that the stronger guarantees of Section II do not apply) and the equivalent of $\mathbf{F}_\kappa$, see (4), is still ill-conditioned. Nonetheless, since SR3 gives a primal-only method, it is straightforward to accelerate using FISTA [8]. In Fig. 7, we see that this accelerated method converges much more rapidly to the minimum loss, giving a significantly better algorithm for TV deblurring. The FISTA algorithm for SR3 TV is detailed in Algorithm 4.

We do not compare the support recovery of the two formulations, (24) and (25), because the original signal does not have a truly sparse discrete gradient. The recovered signals for either formulation have comparable signal-to-noise ratios (SNR), approximately 26.10 for SR3 and 26.03 for standard TV (these numbers vary quite a bit based on parameter choice and maximum number of iterations).

Analysis: We can further analyze SR3 for the specific $\mathbf{C}$ used in the TV denoising problem in order to understand the mediocre performance of unaccelerated SR3. Setting $\boldsymbol{x} = \mathrm{vec}(\mathbf{X})$, we have

$$\mathbf{A} * \mathbf{X} = \mathcal{F}^{-1}\,\mathrm{Diag}(\hat{\boldsymbol{c}})\mathcal{F}\boldsymbol{x}, \quad \mathbf{D}_x\mathbf{X} = \mathcal{F}^{-1}\,\mathrm{Diag}(\hat{\boldsymbol{d}}_x)\mathcal{F}\boldsymbol{x},$$
$$\mathbf{D}_y\mathbf{X} = \mathcal{F}^{-1}\,\mathrm{Diag}(\hat{\boldsymbol{d}}_y)\mathcal{F}\boldsymbol{x}$$

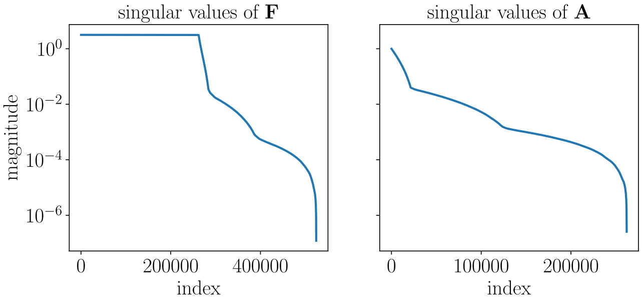


FIGURE 8. **Singular values (ordered by magnitude) of $\mathbf{F}_\kappa$ (left panel) and A (right panel) in the TV example.**

where $\mathcal{F}\boldsymbol{x}$ corresponds to taking a 2D Fourier transform, i.e. of $\mathcal{F}\boldsymbol{x} = \mathrm{vec}(\mathcal{F}^{(2\mathrm{d})}\mathbf{X})$. Then, $\mathbf{F}_\kappa$ can be written as

$$\begin{bmatrix} \kappa\mathcal{F}^{-1}\,\mathrm{Diag}(\hat{\boldsymbol{c}})\mathbf{H}_\kappa^{-1}\begin{bmatrix}\mathrm{Diag}(\hat{\boldsymbol{d}}_x) & \mathrm{Diag}(\hat{\boldsymbol{d}}_y)\end{bmatrix}\mathcal{F} \\ \sqrt{\kappa}\mathcal{F}^{-1}\left(\mathbf{I}-\kappa\begin{bmatrix}\mathrm{Diag}(\hat{\boldsymbol{d}}_x) \\ \mathrm{Diag}(\hat{\boldsymbol{d}}_y)\end{bmatrix}\mathbf{H}_\kappa^{-1}\begin{bmatrix}\mathrm{Diag}(\hat{\boldsymbol{d}}_x) & \mathrm{Diag}(\hat{\boldsymbol{d}}_y)\end{bmatrix}\right)\mathcal{F}\end{bmatrix},$$

where

$$\mathbf{H}_\kappa = \mathcal{F}^{-1}\,\mathrm{Diag}(\hat{\boldsymbol{c}} \odot \hat{\boldsymbol{c}} + \kappa\hat{\boldsymbol{d}}_x \odot \hat{\boldsymbol{d}}_x + \kappa\hat{\boldsymbol{d}}_y \odot \hat{\boldsymbol{d}}_x)\mathcal{F},$$

and $\odot$ is element-wise multiplication. The SR3 formulation (25) reduces to

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\mathbf{F}_\kappa\boldsymbol{w} - \boldsymbol{g}_\kappa\|^2 + \lambda\|\boldsymbol{w}\|_1,$$

with $\mathbf{F}_\kappa$ and $\boldsymbol{g}_\kappa$ as above, and $\boldsymbol{w} = \mathrm{vec}\left(\circ\sqrt{\mathbf{W}_x + \mathbf{W}_y}\right)$, where $\circ\sqrt{A}$ and A denote element-wise square root and squaring operations, respectively.

Setting $\hat{\boldsymbol{h}} = \hat{\boldsymbol{c}} \odot \hat{\boldsymbol{c}} + \kappa\hat{\boldsymbol{d}}_x \odot \hat{\boldsymbol{d}}_x + \kappa\hat{\boldsymbol{d}}_y \odot \hat{\boldsymbol{d}}_x$, we have

$$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa = \mathcal{F}^{-1}\mathcal{A}_\kappa\mathcal{F},$$

with $\mathcal{A}_\kappa$ given by

$$\begin{bmatrix} \kappa\mathbf{I}-\kappa^2\,\mathrm{Diag}(\hat{\boldsymbol{d}}_x\odot\hat{\boldsymbol{h}}^{-1}\odot\hat{\boldsymbol{d}}_x) & -\kappa^2\,\mathrm{Diag}(\hat{\boldsymbol{d}}_x\odot\hat{\boldsymbol{h}}^{-1}\odot\hat{\boldsymbol{d}}_y) \\ -\kappa^2\,\mathrm{Diag}(\hat{\boldsymbol{d}}_y\odot\hat{\boldsymbol{h}}^{-1}\odot\hat{\boldsymbol{d}}_x) & \kappa\mathbf{I}-\kappa^2\,\mathrm{Diag}(\hat{\boldsymbol{d}}_y\odot\hat{\boldsymbol{h}}^{-1}\odot\hat{\boldsymbol{d}}_y) \end{bmatrix}.$$

$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa$ is a 2×2 block system of diagonal matrices, so we can efficiently compute its eigenvalues, thereby obtaining the singular values of $\mathbf{F}_\kappa$. In Figure 8, we plot the spectrum of $\mathbf{F}_\kappa$. Half of the singular values are exactly $\sqrt{\kappa}$, and the other half drop rapidly to 0. This spectral property is responsible for the slow sublinear convergence rate of SR3. Because of the special structure of the $\mathbf{C}$ matrix, $\mathbf{F}_\kappa$ does not improve conditioning as in the LASSO example, where $\mathbf{C} = \mathbf{I}$. The SR3 formulation still makes it easy to apply the FISTA algorithm to the reduced problem (5), improving the convergence rates.

C. SR3 FOR EXACT DERIVATIVES

TV regularizers are often used in physical settings, where the position and the magnitude of the non-zero values for the derivative are of interest. In this numerical example, we use synthetic data to illustrate the efficacy of SR3 for such problems. In particular, we demonstrate that the use of nonconvex regularizers can improve performance.

Problem Setup: Consider a piecewise constant step function with dimension $\boldsymbol{x}_t \in \mathbb{R}^{500}$ and values from -2 to 2, see the first row of Figure 9 for a sample plot. We take 100 random measurements $\boldsymbol{b} = \mathbf{A}\boldsymbol{x}_t + \sigma\boldsymbol{\epsilon}$ of the signal, where the elements of $\mathbf{A}$ and $\boldsymbol{\epsilon}$ are i.i.d. standard Gaussian, and we choose a noise level of $\sigma = 1$.

To recover the signal, we solve the SR3 formulation

$$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \lambda R(\boldsymbol{w}) + \frac{1}{2}\|\boldsymbol{w} - \mathbf{C}\boldsymbol{x}\|^2,$$

where R is chosen to be $\|\cdot\|_0$ or $\|\cdot\|_1$, and $\mathbf{C}$ is the appropriate forward difference matrix. We want to recover the signal $\boldsymbol{x}_t$ and obtain an estimate of the discrete derivative using $\boldsymbol{w}$.

Parameter Selection: We set $\kappa = 1$ and choose λ by cross-validation. We set $\lambda = 0.07$ when $R = \ell_1$ and $\lambda = 0.007$ when $R = \ell_0$.

Results: Results are shown in Figure 9, with the first row showing the recovered signals (red dashed line and green dot-dashed line) vs. true signal (blue solid line) and the second row showing the estimated signal derivative $\boldsymbol{w}$.

If we explicitly use the fact that our signal is a step function, it is easy to recover an accurate approximation of the signal using both $\boldsymbol{x}$ and $\boldsymbol{w}$. We define groups of indices corresponding to contiguous sequences for which $w_i = 0$. For such contiguous groups, we set the value of the recovered signal to be the mean of the x_i values. Ideally, there should be five such groups. In order to recover the signal, we need good group identification (positions of nonzeros in $\boldsymbol{w}$) and an unbiased estimation for signal $\boldsymbol{x}$. From the red dash line in the first row of Figure 9, we can see that both ℓ_0 and ℓ_1 reasonably achieve this goal using the grouping procedure.

However, such an explicit assumption on the structure of the signal may not be appropriate in more complicated applications. A more generic approach would "invert" $\mathbf{C}$ (discrete integration in this example) to reconstruct the signal given $\boldsymbol{w}$. From the second row of Figure 9 we see that ℓ_0-TV obtains a better unbiased estimation of the magnitude of the derivative compared to ℓ_1-TV; accordingly, the signal reconstructed by integration is more faithful using the ℓ_0-style regularizatoin.

D. SR3 FOR MATRIX COMPLETION

Analogous to sparsity in compressed sensing, low-rank structure has been used to solve a variety of matrix completion problems, including the famous Netflix Prize problem, as well as in control, system identification, signal processing [55], combinatorial optimization [12], [43], and seismic data interpolation/denoising [3], [39].

We compare classic rank penalty approaches using the nuclear norm (see e.g. [43]) to the SR3 approach on a seismic interpolation example. Seismic data interpolation is crucial for accurate inversion and imaging procedures such as full-waveform inversion [52], reverse-time migration [6] and multiple removal methods [51]. Dense acquisition is prohibitively expensive in these applications, motivating reduction in seismic measurements. On the other hand, using subsampled sources and receivers without interpolation gives unwanted imaging artifacts. The main goal is to simultaneously sample and compress a signal using optimization to replace dense acquisition, thus enabling a range of applications in seismic data processing at a fraction of the cost.

Problem Setup: We use a real seismic line from the Gulf of Suez. The signal is stored in a 401 × 401 complex matrix, indexed by source/receiver, see the left plot of Fig. 10. Fully sampled seismic data has a fast decay of singular values, while sub-sampling breaks this decay [3]. A convex

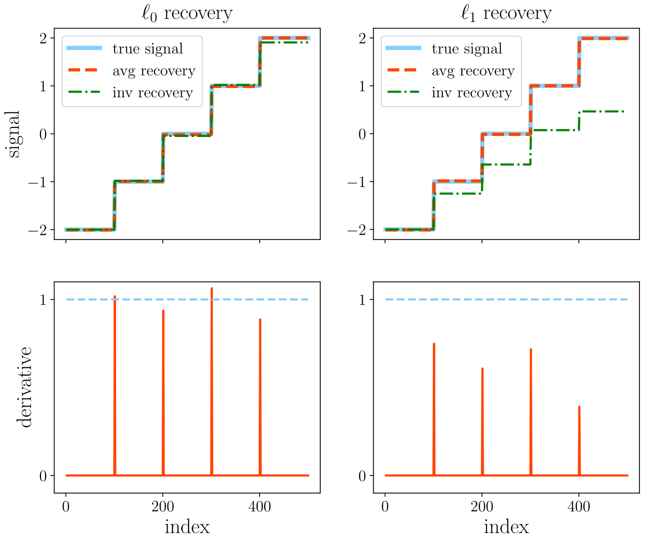


FIGURE 9. SR3 TV regularization result on synthetic data. The first row plots the averaging recovery signal (dashed red line), integrating recovery signal (dot dashed green line) and the true signal (solid blue line). Second row plots the discretized derivative (solid red line) and true magnitude (dashed blue line). First column contain the results come from ℓ_0 regularization, second column is from ℓ_1.

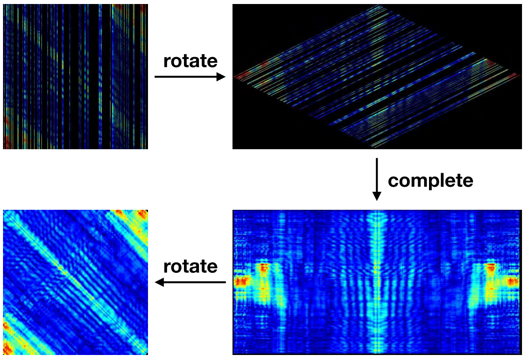


FIGURE 10. Interpolating a frequency slice from the Gulf of Suez dataset. Clockwise we see subsampled data in the source-receiver domain; transformation of the data to the midpont-offset domain, interpolation, and inverse transform back to the source/receiver domain.

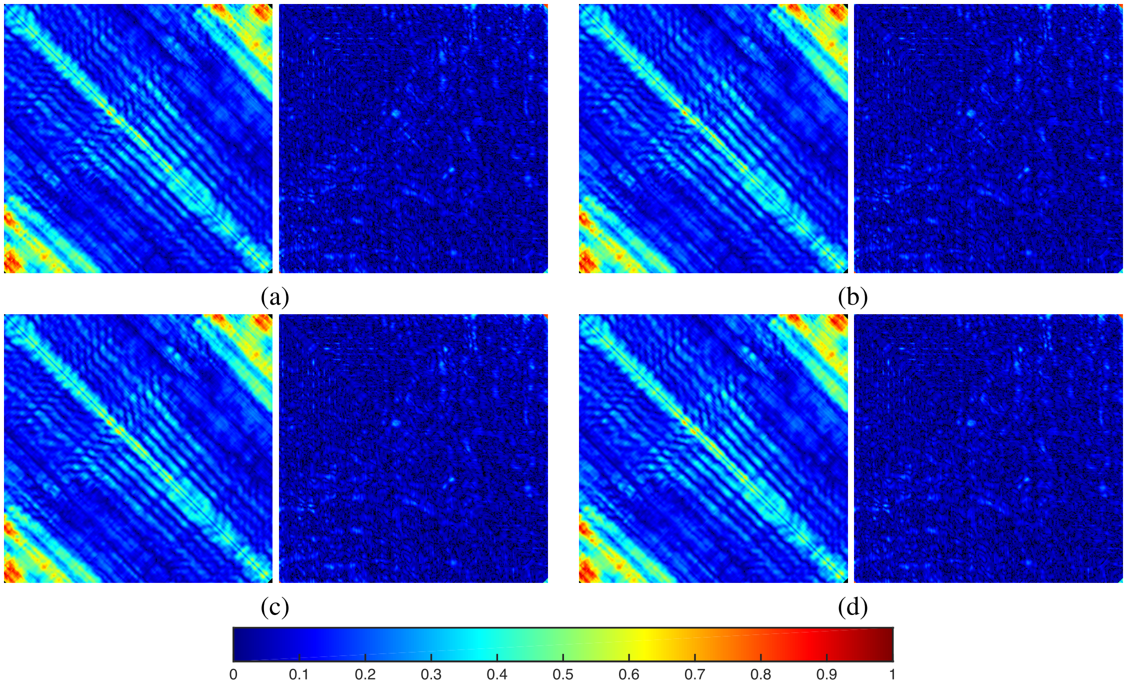


FIGURE 11. Result comparison SR3 vs. classic low rank regression. In each subplot, we show the recovered signal matrix (left) and the difference between recovered the true signal (right). The corresponding SNR is provided. (a), (b) plot the the results of SR3 with ℓ_0 and ℓ_1 regularizers. (c), (d) plot the results of classic formulation with ℓ_0 and ℓ_1 regularizers.

formulation for matrix completion with nuclear norm is given by [43]

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{A}(\mathbf{X})-\mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{X})) \tag{26}$$

where $\mathcal{A}$ maps $\mathbf{X}$ to data $\mathbf{D}$, and $R(\cdot) = \|\cdot\|_1$ penalizes rank.

The SR3 model relaxes (28) to obtain the formulation

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathcal{A}(\mathbf{X})-\mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{W})) + \frac{\kappa}{2}\|\mathbf{W}-\mathbf{X}\|_F^2. \tag{27}$$

To find $\mathbf{X}(\mathbf{W})$, the minimizer of (29) with respect to $\mathbf{X}$, we solve a least squares problem. The $\mathbf{W}$ update requires thresholding the singular values of $\mathbf{X}(\mathbf{W})$.

We compare the results from four formulations, SR3 ℓ_0, SR3 ℓ_1, classic ℓ_0 and classic ℓ_1, i.e. the equations

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{A}(\mathbf{X})-\mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{X})) \tag{28}$$

and

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathcal{A}(\mathbf{X})-\mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{W})) + \frac{\kappa}{2}\|\mathbf{W}-\mathbf{X}\|_F^2, \tag{29}$$

where R can be either ℓ_1 or ℓ_0. To generate figures from SR3 solutions, we look at the signal matrix $\mathbf{X}$ rather than the auxiliary matrix $\mathbf{W}$, since we want the interpolated result rather than a support estimate, as in the compressive sensing examples.

In Figure 10, 85% of the data is missing. We arrange the frequency slice into a 401 × 401 matrix, and then transform the data into the midpoint-offset domain following [3], with $m = \frac{1}{2}(s+r)$ and $h = \frac{1}{2}(s-r)$, increasing the dimension to 401 × 801. We then solve (29) to interpolate the slice, and compare with the original to get a signal-to-noise ratio (SNR) of 9.7 (last panel in Fig. (10)). The SNR obtained by solving (28) is 9.2.

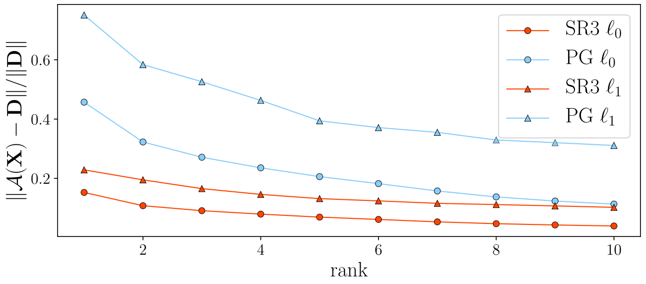


FIGURE 12. Pareto frontiers (best fit achievable for each rank) for (28) with $R = \ell_1$, $R = \ell_0$, and for corresponding SR3 formulations (29), describing the best fits of observed values achievable for a given rank (obtained across regularizers for the four formulations). ℓ_0 formulations are more efficient than those with ℓ_1, and SR3 formulations (29) are more efficient classic formulations (28).

Parameter Selection: We choose $\kappa = 0.5$ for all the experiments and do a cross validation for λ. When $R = \ell_1$, we range λ from 5 to 8; when $R = \ell_0$, we range λ from 200 to 400.

Results: Results are shown in Figures 11 and 12. The relative quality of the images is hard to compare with the naked eye, so we compute the Signal to Noise Ratio (SNR) with respect to the original (fully sampled) data to present a comparison. SR3 fits original data better than the solution of (28), obtaining a maximum SNR of 12.6, see Figure 11.

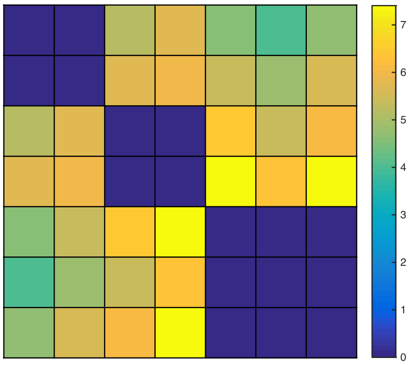


FIGURE 13. Pairwise distance between all decision variables of different tasks obtained by SR3.

We also generate Pareto curves for the four approaches, plotting achievable misfit on the observed data against the ranks of the solutions. Pareto curves for ℓ_0 formulations lie below those of ℓ_1 formulations, i.e. using the 0-norm allows better data fitting for a given rank, and equivalently a lower rank at a particular error level, see Figure 12. The Pareto curves obtained using the SR3 approach are lower still, through the relaxation.

E. SR3 FOR GROUP SPARSITY

Group sparsity is a composite regularizer used in multi-task learning with under-determined tasks. Consider a set of under-determined linear systems,

$$\boldsymbol{b}_i = \mathbf{A}_i\boldsymbol{x}_i + \sigma\boldsymbol{\epsilon}_i, \quad i = 1, \dots, k,$$

where $\mathbf{A}_i \in \mathbb{R}^{m_i \times n}$ and $m_i < n$. If we assume a priori that some of these systems might share the same solution vector, we can formulate the problem of recovering the $\boldsymbol{x}_i$ as

$$\min_{\boldsymbol{x}_i} \frac{1}{2}\sum_{i=1}^{k} \|\mathbf{A}_i\boldsymbol{x}_i - \boldsymbol{b}_i\|_2^2 + \lambda \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$$

where the ℓ_2 norm promotes sparsity of the differences $\boldsymbol{x}_i - \boldsymbol{x}_j$ (or, equivalently, encourages redundancy in the $\boldsymbol{x}_i$). To write the objective in a compact way, set

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_k \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_k \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_k \end{bmatrix}.$$

We can then re-write the optimization problem as

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \|\mathbf{D}_{ij}\boldsymbol{x}\|_2,$$

where $\mathbf{D}_{ij}\boldsymbol{x}$ gives the pairwise differences between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. There is no simple primal algorithm for this objective, as $\|\cdot\|_2$ is not smooth and there is no efficient prox operation for the composition of $\|\cdot\|_2$ with the mapping $\mathbf{D}$.

Applying the SR3 approach, we introduce the variables $\boldsymbol{w}_{ij}$ to approximate $\mathbf{D}_{ij}\boldsymbol{x}$ and obtain

$$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \|\boldsymbol{w}_{ij}\|_2 + \frac{\kappa}{2}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \|\boldsymbol{w}_{ij} - \mathbf{D}_{ij}\boldsymbol{x}\|_2^2.$$

Problem Setup: We set up a synthetic problem with $n = 200$, $m_i = 150$, and $k = 7$. The $\mathbf{A}_i$ are random Gaussian matrices and we group the true underlying signal as follows:

$$\boldsymbol{x}_1 = \boldsymbol{x}_2, \quad \boldsymbol{x}_3 = \boldsymbol{x}_4, \quad \boldsymbol{x}_5 = \boldsymbol{x}_6 = \boldsymbol{x}_7$$

where the generators are sampled form a Gaussian distribution. We set the noise level to $\sigma = 0.1$.

Parameter Selection: We select optimization parameters to be $\lambda = 10$ and $\kappa = 1$.

Results: The pairwise distance of the result is shown in Figure 13. The groups have been successfully recovered. If we directly use the $\boldsymbol{x}$ from the SR3 solution, we obtain 47% relative error. However, using the pattern discovered by $\boldsymbol{w}$ to regroup the least square problems, namely combine $\mathbf{A}_1, \mathbf{A}_2$ and $\boldsymbol{b}_1, \boldsymbol{b}_2$ to solve for the first group of variables, $\boldsymbol{x}_1 = \boldsymbol{x}_2$, and so on, we improve the result significantly to 1% relative error (which is essentially optimal given the noise).

IV. DISCUSSION AND OUTLOOK

Sparsity promoting regularization of regression problems continues to play a critical role in obtaining actionable and interpretable models from data. Further, the robustness, computational efficiency, and generalizability of such algorithms is required for them to have the potential for broad applicability across the data sciences. The SR3 algorithm developed here satisfies all of these important criteria and provides a broadly applicable, simple architecture that is better than state-of-the-art methods for compressed sensing, matrix completion, LASSO, TV regularization, and group sparsity. Critical to its success is the relaxation that splits sparsity and accuracy requirements.

The SR3 approach introduces an additional relaxation parameter. In the empirical results presented here, we did not vary κ significantly, showing that for many problems, choosing $\kappa \approx 1$ can improve over the state of the art. The presence of κ affects the regularization parameter λ, which must be tuned even if a good λ is known for the original formulation. Significant improvements can be achieved by choices of the pair (κ, λ); we recommend using cross-validation, and leave automatic strategies for parameter tuning to future work.

The success of the relaxed formulation suggests broader applicability of SR3. This paper focused on regularized regression, but the method applies more generally to inference over any type of estimator with general statistical assumptions. In particular, we can consider the general optimization problem associated with nonlinear functions, such as the training of neural networks, optimizing over a set of supervised input-output responses that are given by a nonlinear function $f(\cdot)$ with constraints. The relaxed formulation of (2) generalizes to

$$\min_{\boldsymbol{x},\boldsymbol{w}} f(\mathbf{A}, \boldsymbol{x}, \boldsymbol{b}) + \lambda R(\boldsymbol{w}) + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x} - \boldsymbol{w}\|^2. \tag{30}$$

Accurate and sparse solutions for such neural network architectures can be more readily generalizable, analogous with how SR3 helps to achieve robust variable selection in sparse linear models. The application to neural networks is beyond the scope of the current manuscript, but the architecture proposed has great potential for broader applicability.

APPENDIX

We review necessary preliminaries from the optimization literature, and then present a series of theoretical results that explain some of the properties of SR3 solutions and characterize convergence of the proposed algorithms.

MATHEMATICAL PRELIMINARIES

Before analyzing SR3, we give some basic results from the non-smooth optimization literature.

SUBDIFFERENTIAL AND OPTIMALITY

In this paper, we work with nonsmooth functions, both convex and nonconvex. Given a convex nonsmooth function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite, the *subdifferential* of f at $\bar{x}$, denoted $\partial f(\bar{x})$, is the set of all vectors v satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle \quad \forall\, x.$$

The classic necessary stationarity condition $0 \in \partial f(\bar{x})$ implies $f(x) \geq f(\bar{x})$ for all x, i.e. global optimality. The definition of subdifferential must be amended for the general nonconvex case. Given an arbitrary function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and a point $\bar{x}$ with $f(\bar{x})$ finite, the *Fréchet subdifferential* of f at $\bar{x}$, denoted $\hat{\partial} f(\bar{x})$, is the set of all vectors v satisfying

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o(\|x - \bar{x}\|) \quad \text{as } x \to \bar{x}.$$

Thus the inclusion $v \in \hat{\partial} f(\bar{x})$ holds precisely when the affine function $x \mapsto f(\bar{x}) + \langle v, x - \bar{x}\rangle$ underestimates f up to first-order near $\bar{x}$. In general, the limit of Fréchet subgradients $v_i \in \hat{\partial} f(x_i)$, along a sequence $x_i \to \bar{x}$, may not be a Fréchet subgradient at the limiting point $\bar{x}$. Therefore, one formally enlarges the Fréchet subdifferential and defines the *limiting subdifferential* of f at $\bar{x}$, denoted $\partial f(\bar{x})$, to consist of all vectors v for which there exist sequences x_i and v_i, satisfying $v_i \in \partial f(x_i)$ and $(x_i, f(x_i), v_i) \to (\bar{x}, f(\bar{x}), v)$. In this general setting, the condition $0 \in \partial f(\bar{x})$ is necessary but not sufficient. However, stationary points are the best we can hope to find using iterative methods, and distance to stationarity serves as a way to detect convergence and analyze algorithms. In particular, we design and analyze algorithms that find the stationary points of (1) and (5), which are defined below, for both convex and nonconvex regularizers $R(\cdot)$.

Definition 1 (Stationarity): We call $\hat{\boldsymbol{x}}$ the stationary point of (1) *if,*

$$\mathbf{0} \in \mathbf{A}^\top(\mathbf{A}\hat{\boldsymbol{x}} - \boldsymbol{b}) + \lambda \mathbf{C}^\top \partial R(\hat{\boldsymbol{x}}).$$

And $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{w}})$ *the stationary point of* (5) *if,*

$$\mathbf{0} = \mathbf{A}^\top(\mathbf{A}\hat{\boldsymbol{x}} - \boldsymbol{b}) + \kappa \mathbf{C}^\top(\mathbf{C}\hat{\boldsymbol{x}} - \hat{\boldsymbol{w}}),$$
$$\mathbf{0} \in \lambda \partial R(\hat{\boldsymbol{w}}) + \kappa(\hat{\boldsymbol{w}} - \mathbf{C}\hat{\boldsymbol{x}}).$$

MOREAU ENVOLOPE AND PROX OPERATORS

For any function f and real $\eta > 0$, the *Moreau envelope* and the *proximal mapping* are defined by

$$f_\eta(x) := \inf_z \left\{ f(z) + \tfrac{1}{2\eta}\|z - x\|^2 \right\}, \tag{31}$$

$$\text{prox}_{\eta f}(x) := \mathop{arg\,min}_z \left\{ \eta f(z) + \tfrac{1}{2}\|z - x\|^2 \right\}, \tag{32}$$

respectively.

The Moreau envelope has a smoothing effect on convex functions, characterized by the following theorem. Note that a proper function f satisfies that $f > -\infty$ and it takes on a value other than $+\infty$ for some x. A closed function satisfies that $\{x : f(x) \leq \alpha\}$ is a closed set for each $\alpha \in \mathbb{R}$.

Theorem 5 (Regularization Properties of the Envelope): Let $f : \mathbb{R}^n \to \mathbb{R}$ be a proper closed convex function. Then f_η is convex and C^1-smooth with

$$\nabla f_\eta(x) = \tfrac{1}{\eta}(x - \text{prox}_{\eta f}(x)) \quad \textit{and} \quad \text{Lip}(\nabla f_\eta) \leq \tfrac{1}{\eta}.$$

If in addition f is L-Lipschitz, then the envelope $f_\eta(\cdot)$ is L-Lipschitz and satisfies

$$0 \leq f(x) - f_\eta(x) \leq \frac{L^2\eta}{2} \quad \textit{for all } x \in \mathbb{R}^n. \tag{33}$$

Proof: See [44, Th. 2.26]. □

However, when f is not convex, f_η may no longer be smooth as we show in Figure 14 where we use ℓ_0 as an example.

COMMON PROX OPERATORS

The prox operator is useful when designing algorithms that handle non-smooth and non-convex functions. Its calculation is often straightforward when the function f decouples element-wise. To illustrate the idea, we derive proximal mappings for ℓ_1, ℓ_0, ℓ_2^2, and ℓ_2. Many more operators can be found e.g. in [22].

- $f(\cdot) = \|\cdot\|_1$. The ℓ_1 norm is a convex nonsmooth penalty often used to promote sparse solutions in regression problems. We include a derivation of the proximity operator for this problem and the remaining operators have similar derivations.

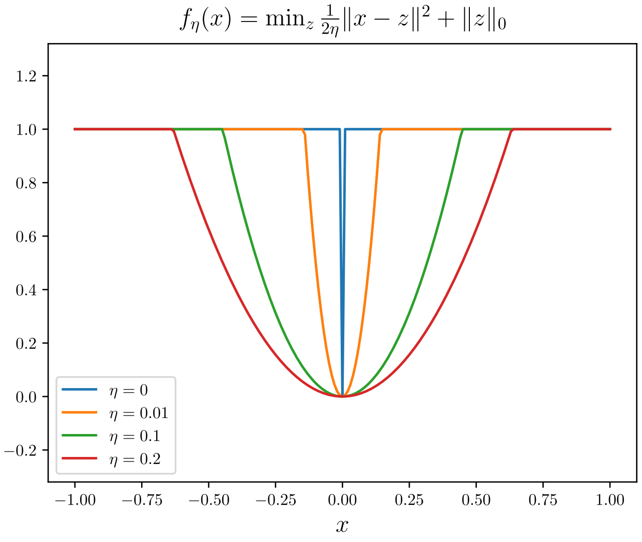


FIGURE 14. Envelope functions indexed by the parameter η, for $f = \|\cdot\|_0$. In contrast to the convex case, here all f_η are nonsmooth and nonconvex.

Lemma 1 (ℓ_1): The prox operator of ℓ_1 is an element-wise soft-thresholding action on the given vector.

$$\boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_1 \ \Rightarrow$$

$$x_i = \begin{cases} y_i - \eta, & y_i > \eta \\ 0, & |y_i| \le \eta \\ y_i + \eta, & y_i < -\eta. \end{cases} \tag{34}$$

Proof: Note that the optimization problem may be written as

$$\begin{aligned} &\underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_1 \\ &\qquad = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\sum_{i=1}^n (x_i - y_i)^2 + \eta|x_i|, \end{aligned} \tag{35}$$

i.e. the problem decouples over the elements of $\boldsymbol{y}$. For each i, the optimization problem has the subdifferential

$$\begin{aligned} &\partial_{x_i}\left(\frac{1}{2}(x_i-y_i)^2 + \eta|x_i|\right) \\ &\qquad = \begin{cases} x_i - y_i + \eta, & x_i > 0 \\ x_i - y_i + \{z : |z| \le \eta\}, & x_i = 0 \\ x_i - y_i - \eta, & x_i < 0. \end{cases} \end{aligned} \tag{36}$$

After checking the possible stationary points given this formula for the subdifferential, it is simple to derive (34). □

- $f(\cdot) = \|\cdot\|_0$. The ℓ_0 penalty directly controls the number of non-zeros in the vector instead of penalizing the magnitude of elements as ℓ_1 does. However, it is non-convex and in practice regression formulations with ℓ_0 regularization can be trapped in local minima instead of finding the true support.

Lemma 2 (ℓ_0): The prox operator of ℓ_0 is simple, element-wise hard-thresholding:

$$\boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_0 \ \Rightarrow$$

$$x_i = \begin{cases} y_i, & |y_i| > \sqrt{2\eta} \\ 0, & |y_i| \le \sqrt{2\eta}. \end{cases} \tag{37}$$

Proof: Analogous to the ℓ_1, the prox problem for ℓ_0 can be decoupled across coordinates:

$$\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_0 = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\sum_{i=1}^n (x_i - y_i)^2 + \eta 1_{\{x_i=0\}}.$$

From this formula, it is clear that the only possible solutions for each coordinate are $x_i = 0$ or $x_i = y_i$. The formula (37) follows from checking the conditions for these cases. □

- $f(\cdot) = \frac{1}{2}\|\cdot\|^2$. The ℓ_2^2 penalty can be used as a smooth and convex penalty which biases towards zero. When combined with linear regression, it is commonly known as ridge regression.

Lemma 3 (ℓ_2^2): The prox of ℓ_2^2 is scaling.

$$\boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \frac{\eta}{2}\|\boldsymbol{x}\|^2 = \frac{1}{1+\eta}\boldsymbol{y}.$$

Proof: The proof follows directly from calculus. □

- $f(\cdot) = \|\cdot\|$. The ℓ_2 norm adds a group sparsity prior, i.e. the vector $\boldsymbol{x}$ is biased toward being the zero vector. Often, this penalty is applied to each column of a matrix of variables. Unlike the prox operators above, $\|\cdot\|$ (by design) does not decouple into scalar problems. Fortunately, a closed form solution is easy to obtain.

Lemma 4:

$$\boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\arg\min}\ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\| \ \Rightarrow$$

$$\boldsymbol{x} = \begin{cases} \frac{\|\boldsymbol{y}\|-\eta}{\|\boldsymbol{y}\|}\boldsymbol{y}, & \|\boldsymbol{y}\| > \eta \\ \boldsymbol{0}, & \|\boldsymbol{y}\| \le \eta. \end{cases}$$

Proof: Observe that for any fixed value of $\|\boldsymbol{x}\|$ the objective

$$\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\| \tag{38}$$

is minimized by taking $\boldsymbol{x}$ in the direction of $\boldsymbol{y}$. This reduces the problem to finding the optimal value of $\|\boldsymbol{x}\|$, for which the same reasoning as the ℓ_1 penalty applies. □

PROXIMAL GRADIENT DESCENT

Consider an objective of the form $p(x) = f(x) + g(x)$. Given a step size t, the proximal gradient descent algorithm is as defined in Algorithm 3 [22]. This algorithm has been studied extensively. Among other results, we have

Theorem 6 (Proximal Gradient Descent): Assume $p = f + g$ and both p and g are closed convex functions. Let p^*

Algorithm 5 Proximal Gradient Descent

```
Input: $x_0$, $\eta$
Initialize: $k = 0$
while not converged do
    $k \leftarrow k + 1$
    $x_k \leftarrow \text{prox}_{\eta g}(x_{k-1} - \eta \nabla f(x_{k-1}))$
Output: $x_k$
```

denote the optimal function value and $\boldsymbol{x}^*$ denote the optimal solution.

- *If ∇f is β Lipschitz continuous, then, setting the step size as $1/\beta$, the iterates generated by proximal gradient descent satisfy*

$$p(\boldsymbol{x}^k) - p^* \leq \frac{\beta \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2}{2(k+1)}.$$

- *Furthermore, if p is also α strongly convex, we have,*

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|^2 \leq \left(1 - \frac{\alpha}{\beta}\right)^k \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2.$$

These results are well known; see e.g. [8], [22], [41] and the tutorial [2, Sec. 4.4].

EQUIVALENCE OF ALGORITHMS 1 AND 2.

In this section we show that Algorithm 1 is equivalent with Algorithm 2. The first lends itself to a straightforward analysis, while the second is far easier to implement.

Variable projection is the key to compute the gradient of the smooth part of v in (3). Here we denote,

$$\begin{aligned} f_s(\boldsymbol{x}, \boldsymbol{w}) &= \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x} - \boldsymbol{w}\|^2, \\ v_s(\boldsymbol{w}) &= \min_{\boldsymbol{x}} f_s(\boldsymbol{x}, \boldsymbol{w}) = f_s(\boldsymbol{x}(\boldsymbol{w}), \boldsymbol{w}), \end{aligned}$$

where $\boldsymbol{x}(\boldsymbol{w}) = arg\,min_{\boldsymbol{x}} f_s(\boldsymbol{x}, \boldsymbol{w})$ We have the relation

$$v(\boldsymbol{w}) = v_s(\boldsymbol{w}) + \lambda R(\boldsymbol{w}).$$

Taking the derivative, we get

$$\nabla v_s(\boldsymbol{w}) = \nabla_{\boldsymbol{w}} f_s(\boldsymbol{x}(\boldsymbol{w}), \boldsymbol{w}) + \nabla_{\boldsymbol{x}} f_s \cdot \left.\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{w}}\right|_{\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{w})}$$

The second term vanishes because $f_s(\boldsymbol{x}(\boldsymbol{w}), \boldsymbol{w})$ is optimal with respect to $\boldsymbol{x}$, and we have

$$\nabla v_s(\boldsymbol{w}) = \kappa(\boldsymbol{w} - \mathbf{C}\boldsymbol{x}(\boldsymbol{w})).$$

From Theorem 1, ∇v_s is Lipchitz continuous with constant κ. Then the proximal gradient step can be written as,

$$\begin{aligned} \boldsymbol{x}^{k+1} &= arg\,min_{\boldsymbol{x}}\, f_s(\boldsymbol{x}, \boldsymbol{w}^k) \\ \boldsymbol{w}^{k+1} &= \text{prox}_{\eta\lambda R}(\boldsymbol{w}^k - \eta\kappa(\boldsymbol{w}^k - \mathbf{C}\boldsymbol{x}^{k+1})) \end{aligned}$$

When we choose $\eta = 1/\kappa$, the second equation becomes,

$$\boldsymbol{w}^{k+1} = \text{prox}_{\lambda/\kappa R}(\mathbf{C}\boldsymbol{x}^{k+1})$$

which precisely matches Algorithm 2.

THEORETICAL RESULTS

In the main text, it is demonstrated that SR3 (5) outperforms the standard regression problem (1), achieving faster convergence and obtaining higher quality solutions. Here, we develop some theory to explain the performance of SR3 from the perspective of the relaxed coordinates, $\boldsymbol{w}$. We obtain an explicit formula for the SR3 problem in $\boldsymbol{w}$ alone and then analyze the spectral properties of that new problem, demonstrating that the conditioning of the $\boldsymbol{w}$ problem is greatly improved over that of the original problem. We also obtain a quantitative measure of the distance between the solutions of the original problem and the SR3 relaxation.

SPECTRAL PROPERTIES OF κ

1) PROOF OF THEOREM 1

The first property can be verified by direct calculation. We have

$$\begin{aligned} \mathbf{F}_\kappa^\top \mathbf{F}_\kappa \boldsymbol{w} - \mathbf{F}_\kappa^\top \boldsymbol{g}_\kappa &= (\kappa \mathbf{I} - \kappa^2 \mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top)\boldsymbol{w} - \kappa \mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{A}^\top \boldsymbol{b} \\ &= \kappa \mathbf{H}_\kappa^{-1}[(\mathbf{H}_\kappa - \kappa\mathbf{I})\boldsymbol{w} - \mathbf{A}^\top \boldsymbol{b}] \\ &= \kappa \mathbf{H}_\kappa^{-1}(\mathbf{A}^\top \mathbf{A}\boldsymbol{w} - \mathbf{A}^\top \boldsymbol{b}) \end{aligned}$$

so that $\mathbf{F}_\kappa^\top \mathbf{F}_\kappa \boldsymbol{w} - \mathbf{F}_\kappa^\top \boldsymbol{g}_\kappa = \mathbf{0} \iff \mathbf{A}^\top \mathbf{A}\boldsymbol{w} + \mathbf{A}^\top \boldsymbol{b} = \mathbf{0}$. By simple algebra, we have,

$$\begin{aligned} \mathbf{F}_\kappa^\top \mathbf{F}_\kappa &= \kappa \mathbf{I} - \kappa^2 \mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \\ \sigma_i(\mathbf{F}_\kappa^\top \mathbf{F}_\kappa) &= \kappa - \kappa^2 \sigma_{n-i+1}(\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top). \end{aligned} \tag{39}$$

Since $\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top$ and $\mathbf{F}_\kappa^\top \mathbf{F}_\kappa$ are positive semi-definite matrices, we have $\mathbf{0} \preceq \mathbf{F}_\kappa^\top \mathbf{F}_\kappa \preceq \kappa \mathbf{I}$. Denote the SVD for $\mathbf{C}$ by $\mathbf{C} = \mathbf{U}_c \Sigma_c \mathbf{V}_c^\top$. When $n \geq d$ and $\mathbf{C}$ is full rank, we know Σ_c is invertible and $\mathbf{V}_c$ is orthogonal. Then

$$\begin{aligned} \mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top &= \mathbf{U}_c \Sigma_c \mathbf{V}_c^\top (\mathbf{A}^\top \mathbf{A} + \kappa \mathbf{V}_c \Sigma_c^2 \mathbf{V}_c^\top)^{-1} \mathbf{V}_c \Sigma_c \mathbf{U}_c^\top \\ &= \mathbf{U}_c (\Sigma_c^{-1} \mathbf{V}_c^\top \mathbf{A}^\top \mathbf{A} \mathbf{V}_c \Sigma_c^{-1} + \kappa \mathbf{I})^{-1} \mathbf{U}_c^\top \end{aligned}$$

This gives a lower bound of the spectrum of $\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top$,

$$\begin{aligned} &\sigma_{\min}(\Sigma_c^{-1} \mathbf{V}_c^\top \mathbf{A}^\top \mathbf{A} \mathbf{V}_c \Sigma_c^{-1}) \geq \sigma_{\min}(\mathbf{A}^\top \mathbf{A})/\sigma_{\max}(\mathbf{C}^\top \mathbf{C}) \\ &\quad \Rightarrow \sigma_{\max}(\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top) \leq 1/(\sigma_{\min}(\mathbf{A}^\top \mathbf{A})/\sigma_{\max}(\mathbf{C}^\top \mathbf{C}) + \kappa) \end{aligned}$$

Then we obtain the conclusion,

$$\begin{aligned} \sigma_{\min}(\mathbf{F}_\kappa^\top \mathbf{F}_\kappa) \geq \kappa &- \frac{\kappa^2}{\sigma_{\min}(\mathbf{A}^\top \mathbf{A})/\sigma_{\max}(\mathbf{C}^\top \mathbf{C}) + \kappa} \\ &= \frac{\sigma_{\min}(\mathbf{A}^\top \mathbf{A})/\sigma_{\max}(\mathbf{C}^\top \mathbf{C})}{1 + \sigma_{\min}(\mathbf{A}^\top \mathbf{A})/(\kappa \sigma_{\max}(\mathbf{C}^\top \mathbf{C}))}. \end{aligned}$$

When $\mathbf{C} = \mathbf{I}$, we have that

$$\begin{aligned} \mathbf{F}_\kappa^\top \mathbf{F}_\kappa &= \kappa[\mathbf{I} - \kappa(\mathbf{A}^\top \mathbf{A} + \kappa \mathbf{I})^{-1}] \\ &= \mathbf{A}^\top (\mathbf{I} + \mathbf{A}\mathbf{A}^\top/\kappa)^{-1} \mathbf{A} \end{aligned}$$

Assume $\mathbf{A} \in \mathbb{R}^{m \times n}$ has the singular value decomposition (SVD) $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times m}$, and $\mathbf{V} \in \mathbb{R}^{m \times m}$. We have

$$\mathbf{F}_\kappa^\top \mathbf{F}_\kappa = \mathbf{V}\Sigma^\top (\mathbf{I} + \Sigma\Sigma^\top/\kappa)^{-1} \Sigma \mathbf{V}^\top.$$

Let $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{l\times l}$ denote the reduced diagonal part of Σ, i.e. the top-left $l \times l$ submatrix of Σ with $l = \min(m, n)$. When $m \geq n$, we have

$$\Sigma = \begin{bmatrix}\hat{\boldsymbol{\Sigma}}\\ \mathbf{0}\end{bmatrix}, \quad \mathbf{F}_\kappa^\top \mathbf{F}_\kappa = \mathbf{V}\hat{\boldsymbol{\Sigma}}^\top(\mathbf{I} + \hat{\boldsymbol{\Sigma}}^2/\kappa)^{-1}\hat{\boldsymbol{\Sigma}}\mathbf{V}^\top \tag{40}$$

And when $m < n$,

$$\Sigma = \begin{bmatrix}\hat{\boldsymbol{\Sigma}} & \mathbf{0}\end{bmatrix}, \quad \mathbf{F}_\kappa^\top \mathbf{F}_\kappa = \mathbf{V}\begin{bmatrix}\hat{\boldsymbol{\Sigma}}^\top(\mathbf{I} + \hat{\boldsymbol{\Sigma}}^2/\kappa)^{-1}\hat{\boldsymbol{\Sigma}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^\top \tag{41}$$

(8) and (9) follow immediately.

Note that the function

$$\frac{x}{\sqrt{1 + x^2/a}}$$

is an increasing function of x when $x, a > 0$. Therefore, by (9), we have

$$\sigma_{\max}(\mathbf{F}_\kappa) = \frac{\sigma_{\max}(\mathbf{A})}{\sqrt{1 + \sigma_{\max}(\mathbf{A})^2/\kappa}}$$

and

$$\sigma_{\min}(\mathbf{F}_\kappa) = \frac{\sigma_{\min}(\mathbf{A})}{\sqrt{1 + \sigma_{\min}(\mathbf{A})^2/\kappa}}.$$

(13) follows by the definition of the condition number.

2) PROOF OF THEOREM 2

For the iterates of the proximal gradient method, we have

$$\boldsymbol{x}_{k+1} = \underset{\boldsymbol{x}}{arg\,min}\ \frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \eta\nabla f(\boldsymbol{x}_k))\|^2 + \eta g(\boldsymbol{x})$$

and from the first order optimality condition we have

$$\begin{aligned}\mathbf{0} &\in \boldsymbol{x}_{k+1} - \boldsymbol{x}_k + \eta\nabla f(\boldsymbol{x}_k) + \eta\partial g(\boldsymbol{x}_{k+1})\\ &\Rightarrow \frac{1}{\eta}(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) + \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)\\ &\in \nabla f(\boldsymbol{x}_{k+1}) + \partial g(\boldsymbol{x}_{k+1})\\ &\Rightarrow (\|\mathbf{A}\|_2^2\mathbf{I} - \mathbf{A}^\top\mathbf{A})(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) \in \partial p(\boldsymbol{x}_{k+1}),\end{aligned}$$

which establishes the first statement. Next, consider the following inequality

$$\begin{aligned}p(\boldsymbol{x}_{k+1}) &= \frac{1}{2}\|\mathbf{A}\boldsymbol{x}_{k+1} - \boldsymbol{b}\|^2 + \lambda R(\boldsymbol{x}_{k+1})\\ &= \frac{1}{2}\|\mathbf{A}\boldsymbol{x}_k - \boldsymbol{b} + \mathbf{A}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)\|^2 + \lambda R(\boldsymbol{x}_{k+1})\\ &= \frac{1}{2}\|\mathbf{A}\boldsymbol{x}_k - \boldsymbol{b}\|^2 + \lambda R(\boldsymbol{x}_{k+1})\\ &\quad + \left\langle \mathbf{A}^\top(\mathbf{A}\boldsymbol{x}_k - \boldsymbol{b}), \boldsymbol{x}_{k+1} - \boldsymbol{x}_k\right\rangle\\ &\quad + \frac{1}{2}\|\mathbf{A}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)\|^2\\ &\leq \frac{1}{2}\|\mathbf{A}\boldsymbol{x}_k - \boldsymbol{b}\|^2 + \lambda R(\boldsymbol{x}_k) - \frac{\|\mathbf{A}\|_2^2}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2\\ &\quad + \frac{1}{2}\|\mathbf{A}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)\|^2,\end{aligned}$$

which implies the inequality

$$\begin{aligned}&\left\langle \boldsymbol{x}_k - \boldsymbol{x}_{k+1}, (\|\mathbf{A}\|_2^2\mathbf{I} - \mathbf{A}^\top\mathbf{A})(\boldsymbol{x}_k - \boldsymbol{x}_{k+1})\right\rangle\\ &\quad\leq p(\boldsymbol{x}_k) - p(\boldsymbol{x}_{k+1})\\ &\quad\Rightarrow \|\mathbf{A}\|_2^2\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \leq p(\boldsymbol{x}_k) - p(\boldsymbol{x}_{k+1}).\end{aligned}$$

Setting $\boldsymbol{v}_{k+1} = (\|\mathbf{A}\|_2^2\mathbf{I} - \mathbf{A}^\top\mathbf{A})(\boldsymbol{x}_k - \boldsymbol{x}_{k+1})$, we have $\|\boldsymbol{v}_{k+1}\|^2 \leq \|\mathbf{A}\|_2^4\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \leq \|\mathbf{A}\|_2^2(p(\boldsymbol{x}_k) - p(\boldsymbol{x}_{k+1}))$. After we add up and simplify, we obtain

$$\begin{aligned}\frac{1}{N}\sum_{k=0}^{N-1}\|\boldsymbol{v}_{k+1}\|^2 &\leq \frac{\|\mathbf{A}\|_2^2}{N}(p(\boldsymbol{x}_0) - p(\boldsymbol{x}_N))\\ &\leq \frac{\|\mathbf{A}\|_2^2}{N}(p(\boldsymbol{x}_0) - p^*),\end{aligned}$$

which is the desired convergence result.

3) PROOF OF THEOREM 3

The result is immediate from combining Theorem 2 and Theorem 1.

4) PROOF OF COROLLARY 2

The result is immediate from combining Theorem 2 and Corollary 1.

CHARACTERIZING OPTIMAL SOLUTIONS OF SR3

In this section, we quantify the relation between the solution of (1) and (5) when $\mathbf{C} = \mathbf{I}$. In this analysis, we fix κ as a constant and set $\mathbf{C} = \mathbf{I}$.

Lemma 5 (Optimality Conditions for (1) and (5)): Define the sets

$$\begin{aligned}\mathcal{S}_1(\boldsymbol{x}, \lambda_1) &= \{\mathbf{A}^\top\mathbf{A}\boldsymbol{x} - \mathbf{A}^\top\boldsymbol{b} + \lambda_1\boldsymbol{v}_1 : \boldsymbol{v}_1 \in \partial R(\boldsymbol{x})\}\\ \mathcal{S}_2(\boldsymbol{w}, \lambda_2) &= \{\kappa\mathbf{H}_\kappa^{-1}(\mathbf{A}^\top\mathbf{A}\boldsymbol{w} - \mathbf{A}^\top\boldsymbol{b}) + \lambda_2\boldsymbol{v}_2 : \boldsymbol{v}_2 \in \partial R(\boldsymbol{w})\},\end{aligned}$$

where $\mathbf{H}_\kappa = \mathbf{A}^\top\mathbf{A} + \kappa\mathbf{I}$, *as above. These sets contain the subgradients of* (1) *and* (5)*. If we assume* $\hat{\boldsymbol{x}}$ *and* $\hat{\boldsymbol{w}}$ *are the (stationary) solutions of* (1) *and* (5)*, namely*

$$\mathbf{0} \in \mathcal{S}_1(\hat{\boldsymbol{x}}, \lambda_1), \quad \mathbf{0} \in \mathcal{S}_2(\hat{\boldsymbol{w}}, \lambda_2),$$

then

$$\begin{aligned}&[\mathbf{I} - (\lambda_1/\lambda_2)\kappa\mathbf{H}_\kappa^{-1}](\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}) \in \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1),\\ &[\kappa\mathbf{H}_\kappa^{-1} - (\lambda_2/\lambda_1)\mathbf{I}](\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{x}} - \mathbf{A}^\top\boldsymbol{b}) \in \mathcal{S}_2(\hat{\boldsymbol{x}}, \lambda_2).\end{aligned}$$

Proof: As $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{w}}$ are the (stationary) solutions of (1) and (5), we have

$$\begin{aligned}&\exists\boldsymbol{v}_1 \in \partial R(\hat{\boldsymbol{x}}), \quad \lambda_1\boldsymbol{v}_1 = -(\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{x}} - \mathbf{A}^\top\boldsymbol{b}),\\ &\exists\boldsymbol{v}_2 \in \partial R(\hat{\boldsymbol{w}}), \quad \lambda_2\boldsymbol{v}_2 = -\kappa\mathbf{H}_\kappa^{-1}(\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}).\end{aligned}$$

Then,

$$\begin{aligned}&\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b} + \lambda_1\boldsymbol{v}_2 \in \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)\\ &\quad\Rightarrow [\mathbf{I} - (\lambda_1/\lambda_2)\kappa\mathbf{H}_\kappa^{-1}](\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}) \in \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1),\\ &\kappa\mathbf{H}_\kappa^{-1}(\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{x}} - \mathbf{A}^\top\boldsymbol{b}) + \lambda_2\boldsymbol{v}_1 \in \mathcal{S}_2(\hat{\boldsymbol{x}}, \lambda_2)\\ &\quad\Rightarrow [\kappa\mathbf{H}_\kappa^{-1} - (\lambda_2/\lambda_1)\mathbf{I}](\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{x}} - \mathbf{A}^\top\boldsymbol{b}) \in \mathcal{S}_2(\hat{\boldsymbol{x}}, \lambda_2).\end{aligned}$$

□

5) PROOF OF THEOREM 4

Using the definitions of Lemma 5, we have

$$\begin{aligned}
&\mathrm{dist}(\mathbf{0}, \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)) \\
&\quad \le \frac{1}{\hat{\tau}}\|(\hat{\tau}\mathbf{I} - \kappa \mathbf{H}_\kappa^{-1})(\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b})\| \\
&\quad = \frac{1}{\hat{\tau}}\|\hat{\tau}\mathbf{I} - \kappa \mathbf{H}_\kappa^{-1}\|_2\|\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}\| \\
&\quad = \frac{1}{\hat{\tau}}\|\hat{\tau}\mathbf{1} - \kappa \sigma(\mathbf{H}_\kappa^{-1})\|_\infty\|\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}\| \\
&\quad = \frac{\sigma_{\max}(\mathbf{H}_\kappa) - \sigma_{\min}(\mathbf{H}_\kappa)}{\sigma_{\max}(\mathbf{H}_\kappa) + \sigma_{\min}(\mathbf{H}_\kappa)}\|\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}\| \\
&\quad = \frac{\sigma_{\max}(\mathbf{A})^2 - \sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2 + \sigma_{\min}(\mathbf{A})^2 + 2\kappa}\|\mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b}\|.
\end{aligned}$$

If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{w}}$, then $\boldsymbol{r} = \mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{w}} - \mathbf{A}^\top\boldsymbol{b} = \mathbf{A}^\top\mathbf{A}\hat{\boldsymbol{x}} - \mathbf{A}^\top\boldsymbol{b}$ is in the null space of $\tau\mathbf{I} - \kappa\mathbf{H}_\kappa^{-1}$, where $\tau = \lambda_2/\lambda_1$. This establishes a connection between λ_1 and λ_2. For instance, we have the following result. In the case that $\mathbf{A}$ has orthogonal rows or columns, theorem 4 provides some explicit bounds on the distance between these solutions.

Corollary 3 If $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$, *then* $\mathrm{dist}(\mathbf{0}, \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)) = 0$, *i.e.* $\hat{\boldsymbol{w}}$ *is the stationary point of* (1). *If* $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$, *then* $\mathrm{dist}(\mathbf{0}, \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)) \le 1/(1 + 2\kappa)$.

Proof: The formula for $\mathbf{H}_\kappa$ simplifies under these assumptions. When $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$, we have $\mathbf{H}_\kappa = (1 + \kappa)\mathbf{I}$ and $\sigma_{\max}(\mathbf{H}_\kappa) = \sigma_{\min}(\mathbf{H}_\kappa) = 1 + \kappa$. When $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$, we have $\sigma_{\max}(\mathbf{H}_\kappa) = 1+\kappa$ and $\sigma_{\min}(\mathbf{H}_\kappa) = \kappa$. Theorem 4 then implies the result. □

IMPLEMENTATION OF ℓ_Q PROXIMAL OPERATOR

Here we summarize our implementation. The first and second derivatives are given by

$$\begin{aligned}
f'_{\alpha,p}(x; z) &= \frac{1}{\alpha}(x - |z|) + px^{p-1}, \\
f''_{\alpha,p}(x; z) &= \frac{1}{\alpha} + p(p-1)x^{p-2}.
\end{aligned} \tag{42}$$

The point $\tilde{x} = \sqrt[p-2]{-1/(\alpha p(p-1))}$ is the only inflection point of $f_{\alpha,p}$, with $f''_{\alpha,p}(x) < 0$ for $0 \le x < \tilde{x}$, and $f''_{\alpha,p}(x; z) > 0$ when $x > \tilde{x}$.

- If $f'_{\alpha,p}(\tilde{x}; z) \ge 0$, we have $f'_{\alpha,p}(x; z) \ge 0$, for all $x \ge 0$. Then $\mathop{arg\,min}_{x\ge 0} f_{\alpha,p}(x; z) = 0$.
- If $f'_{\alpha,p}(\tilde{x}; z) < 0$, one local min $\bar{x} \in (\tilde{x}, |z|)$ exists, and we can use Newton's method to find it. Then we compare the values at 0 and $\bar{x}$, obtaining

$$\mathop{arg\,min}_{x\ge 0}\; f_{\alpha,p}(x; z) = \begin{cases} 0, & f_{\alpha,p}(0; z) \le f_{\alpha,p}(\bar{x}; z) \\ \bar{x}, & f_{\alpha,p}(0; z) > f_{\alpha,p}(\bar{x}; z). \end{cases}$$

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Rajiv Kumar for his help with the seismic example in Section III-D.

REFERENCES

[1] F. Acker and M.-A. Prestel, "Convergence d'un schéma de minimisation alternée," *Annales Faculté Sci. Toulouse, Mathématiques*, vol. 2, no. 1, pp. 1–9, 1980.

[2] A. Aravkin, J. V. Burke, L. Ljung, A. Lozano, and G. Pillonetto, "Generalized Kalman smoothing: Modeling and algorithms," *Automatica*, vol. 86, pp. 63–86, Dec. 2017.

[3] A. Aravkin, R. Kumar, H. Mansour, B. Recht, and F. J. Herrmann, "Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation," *SIAM J. Sci. Comput.*, vol. 36, no. 5, pp. S237–S266, 2014.

[4] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[5] H. Bauschke, P. Combettes, and D. Noll, "Joint minimization with alternating bregman proximity operators," *Pacific J. Optim.*, vol. 2, no. 3, pp. 401–424, 2006.

[6] E. Baysal, D. D. Kosloff, and J. W. Sherwood, "Reverse time migration," *Geophysics*, vol. 48, no. 11, pp. 1514–1524, 1983.

[7] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

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

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[11] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, 2015.

[12] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.

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

[14] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[15] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[16] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[17] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, Mar. 1998.

[18] F. Chen, L. Shen, and B. W. Suter, "Computing the proximity operator of the ℓ_p norm with $0 < p < 1$," *IET Signal Process.*, vol. 10, no. 5, pp. 557–565, 2016.

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

[20] W. Cheney and A. A. Goldstein, "Proximity maps for convex sets," *Proc. Amer. Math. Soc.*, vol. 10, no. 3, pp. 448–450, 1959.

[21] J. F. Claerbout and F. Muir, "Robust modeling with erratic data," *Geophysics*, vol. 38, no. 5, pp. 826–844, 1973.

[22] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2011, pp. 185–212.

[23] A. R. Conn, "Constrained optimization using a nondifferentiable penalty function," *SIAM J. Numer. Anal.*, vol. 10, no. 4, pp. 760–784, 1973.

[24] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philos. Trans. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 367, no. 1906, pp. 4273–4293, 2009.

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

[26] D. Drusvyatskiy and C. Kempton. (2015). "Variational analysis of spectral functions simplified." [Online]. Available: https://arxiv.org/abs/1506.05170

[27] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, p. 947, 2007.

[28] J. Fan and R. Li, ''Variable selection via nonconcave penalized likelihood and its oracle properties,'' *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.

[29] H.-Y. Gao and A. G. Bruce, ''Waveshrink with firm shrinkage,'' *Statistica Sinica*, vol. 7, no. 4, pp. 855–874, 1997.

[30] C. Gauss, ''Theory of the combination of observations which leads to the smallest errors,'' *Gauss Werke*, vol. 4, pp. 1–93, 1821.

[31] C. F. Gauss, ''Theoria motus corporum coelestum,'' *Werke*, 1809.

[32] T. Goldstein and S. Osher, ''The split Bregman method for L1-regularized problems,'' *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[33] I. Goodfellow, Y. Bengio, A. Courville, and F. Bach, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[34] A. E. Hoerl and R. W. Kennard, ''Ridge regression iterative estimation of the biasing parameter,'' *Commun. Statist.-Theory Methods*, vol. 5, no. 1, pp. 77–88, 1976.

[35] J. Huang, J. Wang, F. Zhang, and W. Wang, ''New sufficient conditions of signal recovery with tight frames via l_1-analysis approach,'' *IEEE Access*, vol. 6, pp. 26718–26728, 2018.

[36] M.-J. Lai, Y. Xu, and W. Yin, ''Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization,'' *SIAM J. Numer. Anal.*, vol. 51, no. 2, pp. 927–957, 2013.

[37] A.-M. Legendre, *Nouvelles Méthodes Pour la Détermination des Orbites des Comètes*, F. Didot, Ed. 1805. [Online]. Available: https://archive.org/details/nouvellesmthode00legegoog/page/n9

[38] A. S. Lewis, ''Nonsmooth analysis of eigenvalues,'' *Math. Program.*, vol. 84, no. 1, no. 1, pp. 1–24, 1999.

[39] V. Oropeza and M. Sacchi, ''Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis,'' *Geophysics*, vol. 76, no. 3, pp. V25–V32, 2011.

[40] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, ''An iterative regularization method for total variation-based image restoration,'' *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–489, 2005.

[41] N. Parikh and S. Boyd, ''Proximal algorithms,'' *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[42] T. Pietrzykowski, ''An exact potential method for constrained maxima,'' *SIAM J. Numer. Anal.*, vol. 6, no. 2, pp. 299–304, 1969.

[43] B. Recht, M. Fazel, and P. A. Parrilo, ''Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,'' *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

[44] R. Rockafellar, R. J.-B. Wets, and M. Wets, *Variational Analysis* (Grundlehren der mathematischen Wissenschaften), vol. 317. Berlin, Germany: Springer, 1998.

[45] L. I. Rudin, S. Osher, and E. Fatemi, ''Nonlinear total variation based noise removal algorithms,'' *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[46] C. Shaobing and D. Donoho, ''Basis pursuit,'' in *Proc. 28th Asilomar Conf. Signals, Syst. Comput.*, Oct./Nov. 1994, pp. 41–44.

[47] D. Strong and T. Chan, ''Edge-preserving and scale-dependent properties of total variation regularization,'' *Inverse Problems*, vol. 19, no. 6, p. S165, 2003.

[48] W. Su, M. Bogdan, and E. Candès, ''False discoveries occur early on the lasso path,'' *Ann. Statist.*, vol. 45, no. 5, pp. 2133–2150, 2017.

[49] R. Tibshirani, ''Regression shrinkage and selection via the lasso,'' *J. Roy. Stat. Soc. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[50] A. Tihonov, ''On stability of inverse problems,'' USSR Acad. Sci., Saint Petersburg, Russia, DAN SSSR Rep., 1943, vol. 39, pp. 195–198.

[51] D. J. Verschuur, A. J. Berkhout, and C. P. A. Wapenaar, ''Adaptive surface-related multiple elimination,'' *Geophysics*, vol. 57, no. 9, pp. 1166–1177, 1992.

[52] J. Virieux and S. Operto, ''An overview of full-waveform inversion in exploration geophysics,'' *Geophysics*, vol. 74, no. 6, pp. WCC1–WCC26, 2009.

[53] J. von Neumann, *Functional Operators, The Geometry of Orthogonal Spaces: 2*, vol. 2. Princeton, NJ, USA: Princeton Univ. Press, 1950.

[54] Y. Wang, J. Yang, W. Yin, and Y. Zhang, ''A new alternating minimization algorithm for total variation image reconstruction,'' *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, Aug. 2008.

[55] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, ''Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 156–171, Jan. 2017.

[56] C.-H. Zhang, ''Nearly unbiased variable selection under minimax concave penalty,'' *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, 2010.

[57] P. Zheng and A. Aravkin. (2018). ''Relax-and-split method for nonsmooth nonconvex problems.'' [Online]. Available: https://arxiv.org/abs/1802.02654

Authors' photographs and biographies not available at the time of publication.