Robust Modeling and Algorithm Design for Science and Engineering

Peng Zheng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Aleksandr Aravkin, Chair

Nathan Kutz

Dmitriy Drusvyatskiy

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Robust Modeling and Algorithm Design for Science and Engineering

Peng Zheng

Chair of the Supervisory Committee:
Assistant Professor Aleksandr Aravkin
Department of Applied Mathematics

Efficiently extracting information from data sets is at the core of modern scientific computing and data-driven discovery. Modeling and algorithm design thus become crucial for research in many scientific and engineering domains. We develop formulations that fuse physics-based and data-driven models, use robust statistics to integrate information from noisy sources, and enforce the solution structure to incorporate domain knowledge. These formulations are mathematically challenging, as non-smooth structure and non-convex geometry make algorithm design and analysis difficult. The technical thrust of the research targets these non-convex, non-smooth problems to obtain provably convergent efficient methods.

To help solve fundamental problems in science and engineering, we develop and implement methods in the context of specific applications, including phase retrieval, data decomposition, dynamic inference, and brain imaging. We have developed open source software packages, and shared them with our collaborators and the broader research community via GitHub.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I am very grateful for all the help and support from my family and mentors. I would like to thank my advisor, Professor Aleksandr Aravkin, for introducing me to the world of optimization, patiently guiding me through research difficulties and being a really good friend. His enthusiasm and positive attitude brightened my days during the most challenging period of my PhD. I really enjoyed and appreciated the time we spent together, and I am very excited to continue the friendship and collaborations in the future days.

I am very grateful to Professor Nathan Kutz and Professor Steven Brunton, for all the help and advice they gave. During the second half of my PhD, we collaborated extensively on many interesting topics including sparsity promoting regression and decomposition. They introduced me to many physics domain related problems and gave the optimization tools we developed a purpose. They provided deep intuition and valuable advice for improving tools, writing, and presentations. I am looking forward to continue working with them and learning from them.

I am very thankful for the collaboration with Professor Travis Askham and Dr. Benjamin Erichson. Travis and Ben contributed a lot to the work in Chapter 3, 4 and 5, and I learned a lot from them. I also would like to thank Dr. Aurelie Lozano and Dr. Karthikeyan Natesan Ramamurthy from IBM, for the fun and interesting collaboration experience. Chapter 6 and 7 come from the joint work with them on the topic of computer vision and robust sparse regression.

At the end, I would like thank my parents, as without their support I would never be able to start or finish this degree.

DEDICATION

to my parents, Huilin and Yuxia

Chapter 1

INTRODUCTION

Efficiently extracting information from data sets is at the core of modern scientific computing and data-driven discovery. Modeling and algorithm design thus become crucial for research in many scientific and engineering domains. We develop formulations that fuse physics-based and data-driven models, use robust statistics to integrate information from noisy sources, and enforce the solution structure to incorporate domain knowledge. These formulations are mathematically challenging, as non-smooth structure and non-convex geometry make algorithm design and analysis difficult. The technical thrust of the research targets these non-convex, non-smooth problems to obtain provably convergent efficient methods.

To help solve fundamental problems in science and engineering, we develop and implement methods in the context of specific applications, including phase retrieval, data decomposition, dynamic inference, and brain imaging. We have developed open source software packages, and shared them with our collaborators and the broader research community via GitHub.

In the following, we give a brief introduction of the content and map main body of the thesis. We separate the thesis into methodological contributions (optimization algorithms), new robust models, and numerous applications that make use of the new models and fast methods to fit them.

1.1 Optimization Algorithms

Two key methodological contributions are intertwined with all of the applications presented in this thesis. First is the *relax and split* technique that we developed to solve non-smooth, non-convex problems. This approach makes it possible to solve difficult problems efficiently and elegantly. Second is *variable projection*, a general framework that we have extended and adapted to develop fast algorithms for many problem classes. These contributions are described in more details below.

Relax and split. Many penalties and regularizers used in formulation design are non-smooth and non-convex functions. While they are essential to efficiently capture the structure of the problem and the model, their irregular behavior compared to smooth or convex functions makes design and analysis of the optimization algorithms much more difficult.

We develop the 'relax and split' (RS) framework [383] to solve a general class of non-smooth, nonconvex problems

$$\min_{x} \quad h(Ax) + g(x)$$

where h can be any non-smooth non-convex function while g is smooth and convex. This problem class covers many applications including phase retrieval, semi-supervised learning, stochastic shortest path and protein structure design. The framework provides fast elegant algorithms that simultaneously exploit simplicity of non-convex scalar optimization and efficiency of large-scale linear and convex solvers, and as a result have both better convergence rates and significant computational advantages.

Variable projection. Variable projection (VP), which is also called 'partial minimization', was originally proposed for solving separable non-linear least-square problems. Early work of [166] found success in applications for chemistry, mechanical systems, neural networks and telecommunications [164, 267]. More recent developments extend the problem class to more general settings [39, 17, 16].

VP solves structured optimization problems by partially minimizing over a subset of the variables while iterating over the remaining variables. Consider the objective,

$$\min_{x,\theta} \quad f(x,\theta) + r_1(x) + r_2(\theta),$$

which covers many applications, including robust dynamical mode decomposition and sparse principle component analysis. VP takes advantage of the fact that some of the variables (θ) may be much easier to optimize compared to the remaining variables (x). Partially minimizing over θ reduces the problem to

$$\min_{x} \quad v(x) + r_1(x), \quad \text{s.t.} \;\; v(x) = \min_{\theta} \; f(x,\theta) + r_2(\theta),$$

which we show is better conditioned and converges much faster in practice compared to alternative approaches.

1.2 Robust Models

Many applications struggle with contaminated and noisy data due to a range of causes, including equipment malfunction, or intrinsically high noise in the data collecting mechanism (e.g. survey or census data). Robust models allow high quality inference from these low-fidelity real-world contaminated data sets. We present two types of approaches: development of new robust penalties, and use of trimming to simultaneously detect the outliers as we learn the model.

Robust penalties. In many machine learning and statistical models, we assume errors are additive and follow a Gaussian distribution, largely due to the computational simplicity of the resulting least squares problem. However, this assumption is often violated, as e.g. in the presence of data contamination. Consider a simple regression problem,

$$y_i = \langle a_i, x \rangle + \epsilon_i$$

where $\{y_i, a_i\}$ is a training datum and ϵ_i is the measurement error. Rather than using the Gaussian distribution, we can assume ϵ_i come from a heavy tail distribution, e.g. Huber [195] or Student's T [214]. The resulting problem will then take the form

$$\min_x \quad \sum_{i=1}^{m} f_i(y_i - \langle a_i, x \rangle),$$

where f_i is the negative log likelihood of the error distribution. Doing this reduces the influence of outliers and improves the quality of the resulting solutions. The choice and design of the robust penalties that correspond to heavy tail distributions has a strong impact on the solution. In [384], we demonstrate how customizing the penalty can improve practical performance.

Trimming. Robust penalties have several limitations. Penalty design as described above corresponds to additive error models, in contrast to probabilistic learning models (e.g. SVM, logistic regression, and neural nets); and even for simple regression the relationship between the meta-parameters (e.g. degrees of freedom) and behavior of the corresponding solution must be carefully worked out. These drawbacks are addressed by trimming, originally proposed for the least squares model by [300]. Considering the same regression problem, instead of applying robust penalties, we use auxiliary weights to inform whether each data point is an inlier or outlier, thus controlling its contribution to the overall objective:

$$\min_{x,w} \quad \sum_{i=1}^{m} w_i f_i(y_i - \langle a_i, x \rangle), \quad \text{s.t.} \;\; w_i \in [0,1], \sum_{i=1}^{m} w_i = h.$$

Solving the problem jointly over (x, w) is referred to as 'trimming'. The trimming approach

- completely eliminates the influence of particular data pairs $\{y_i, a_i\}$,
- preserves the original structure of the problem (here least-squares)
- easily extends to more complex models, such as general linear models and neural nets.

We use trimming to develop robust variants of multiple applications throughout the thesis, along with customized algorithms for such applications and more general trimmed formulations. We also develop a new approach to sparse regression based on trimming the regularizer [374].

1.3 Problems and Applications

The main applications in this thesis can be divided into two categories, domain science and machine learning. In the following, we briefly describe each individual application within these areas in more detail. A large part of this thesis was done by engaging closely with domain scientists, learning about specific problems, and then generalizing the methods.

Phase retrieval. Phase retrieval has many variations in crystallography [253, 252], and the two key categories are coherent diffractive imaging [111] and ptychography [244, 112]. In these applications, we shoot coherent X-ray beams at the crystals or other materials, and record the intensity of the diffraction pattern. From the pattern, we want to recover the corresponding phase information which informs material properties. Phase retrieval is formulated as the following optimization problem,

$$\min_{x} \quad \rho(h(Ax) - b)$$

where ρ and h can be chosen as $\|\cdot\|_1$ or $\|\cdot\|_2^2$. Many algorithms have been studied by [145, 146, 160]. Recently, phase retrieval has gained some attention with the work of [78, 127, 133] and [108]. Phase retrieval is challenging due to its non-convex structure, and complications from real experiments, including unknown experimental parameters and noisy data. We apply the relax and split algorithm to solve a simplified version of the problem and show its superior performance compared to other existing algorithms in the field.

Robust dynamical mode decomposition. The dynamic mode decomposition (DMD) is a broadly applicable dimensionality reduction technique that describes time-series dynamic data. It has been used in a variety of fields where the nature of the data can lead to corrupt and noisy measurements. This includes applications ranging from neuroscience to video processing to fluid dynamics.

We propose a framework and a set of algorithms for incorporating robust features (robust penalties and trimming) into the nonlinear 'optimized DMD' framework [21]:

$$\min_{\alpha,B} \quad \sum_{j=1}^{n} w_j \rho(x_j - \Phi(\alpha) b_j), \quad \text{s.t.} \;\; w_j \in [0,1], \sum_{j=1}^{n} w_j = h.$$

We develop a scalable stochastic variable projection algorithm, which allows regularizers and constraints on the objective.

Robust sparse principle component analysis. A wide range of phenomena in the physical, engineering, biological, and social sciences feature rich dynamics that give rise to multiscale structures in both space and time. Remarkably, many of the underlying dynamics of such systems are inherently low-rank, and dimensionality reduction techniques are frequently used to obtain interpretable characterizations of these dynamics. Sparse principal component analysis (SPCA) [387] has emerged as a powerful technique for this purpose, providing improved interpretation of low-rank structures by identifying localized spatial structures in the data and disambiguating between distinct time scales.

We propose a robust and scalable variable projection SPCA algorithm, along with randomized linear algebraic extensions for the large-scale (big data) setting. A new robust SPCA formulation is also able to obtain meaningful sparse components in spite of grossly

corrupted input data, and the approach can include various regularizers, including ℓ_0, ℓ_1, ℓ_2 and their combinations to improve the quality and interpretability of the solution. We provide the theoretical convergence guarantees and demonstrated the exceptional computational efficiency and diagnostic performance of the algorithm using both synthetic and real world data. A open source R library is provided for the use of researchers.

Robust principle component analysis for computer vision. Background-foreground separation is a classical application for robust principle component analysis (RPCA) in computer vision. The idea is that the moving foreground of a still background can be modeled as a sparse noise in a low rank matrix whose columns are the frames from the video. It can be formulated as, [79]

$$\min_{U,V} \quad \frac{1}{2}\|UV^\top + S - Y\|_F^2 + \kappa\|S\|_1 \iff \min_{U,V} \quad \rho_H(UV^\top + S - Y)$$

By partially minimizing S, we obtain the Huber penalty. Equivalently, we can assume the residual distribution is induced by the Huber penalty. This interpretation guides us to a better design for this specific application, and we propose a new penalty we dub 'Tiber':

$$\min_{U,V} \quad \rho_T(UV^\top - Y).$$

Tiber is a huberized student's T penalty that is better aligned with the empirical distribution of the residuals we obtain in the background-foreground application.

Sparse regression. Sparse regression has been instrumental in scientific model discovery, including compressed sensing applications, variable selection, and high-dimensional analysis [329, 117]. Despite tremendous methodological progress over the last decade, 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 non-convex implementations, and (iv) high-fidelity requirements on data.

Here we present two methods, sparse relaxed regularized regression (SR3) [385] and trimmed ℓ_1 method [374], that address many of these issues. Consider the classical sparse regression problem,

$$\min_{x} \quad \frac{1}{2}\|Ax - b\|^2 + R(x).$$

To derive SR3, we apply the relax and split philosophy, and introduce an auxiliary variable w to help carry the burden of the sparsity,

$$\min_{x,w} \quad \frac{1}{2}\|Ax - b\|^2 + R(w) + \frac{1}{2\nu}\|x - w\|^2 \iff \min_{w} \quad \frac{1}{2}\|F_\nu w - g_\nu\|^2 + R(w).$$

By partially minimizing x, we switch the role of w, reframing the problem with w as the only variable. This yields a better conditioned problem that also has better signal recover rate, is

more robust to measurement noise, has better empirical and theoretical convergence rates, and allows us to incorporate a wide class of non-convex sparsity promoting regularizers.

A second complementary line of work improves sparse signal recovery by trimming the *regularizer* rather than the measurements:

$$\min_x \quad \frac{1}{2}\|Ax-b\|^2 + \sum_{i=1}^{n} w_i|x_i|, \quad \text{s.t.} \;\; w_i \in [0,1], \sum_{i=1}^{n} w_i = h.$$

This approach is a new application of trimming and is particularly powerful when the true sparsity level is known. It eliminates the bias introduced by the ℓ_1 norm and helps avoid false positive basis element identifications. In [374], we develop a provable convergent algorithm for this problem and demonstrate its flexibility and strength both theoretically and on several applications, including neural nets.

Chapter 2

RELAX AND SPLIT ALGORITHM

We develop and analyze a new 'relax-and-split' (RS) approach for compositions of separable nonconvex nonsmooth functions with linear maps. RS uses a relaxation technique together with partial minimization, and brings classic techniques including direct factorization, matrix decompositions, and fast iterative methods to bear on nonsmooth nonconvex problems. We also extend the approach to trimmed nonconvex-composite formulations; the resulting Trimmed RS (TRS) can fit models while detecting outliers in the data.

We then test RS and TRS on a diverse set of applications: (1) phase retrieval, (2) stochastic shortest path problems, (3) semi-supervised classification, and (4) new clustering approaches. RS/TRS can be applied to models with very weak functional assumptions, are easy to implement, competitive with existing methods, and enable a new level of modeling formulations to be put forward to address emerging challenges in the mathematical sciences.

2.1 Introduction

Extracting information from large-scale datasets is essential for modern scientific computing and data-driven discovery. Classic techniques such as least squares and direct decompositions (such as the singular value decomposition) demand a prohibitively high degree of data quality, regularity, and homogeneity. Inference in many settings requires robustness to error, enforcement of solution structure, and control of model complexity. These features can be effectively captured using nonsmooth and nonconvex optimization formulations.

In this paper, we consider *nonconvex-composite* problems:

$$\min_{x} \quad f(x) := h(Ax) + g(x), \tag{2.1}$$

where $x \in \mathbb{R}^n$ are decision variables, $A = [a_1, \ldots, a_m]^\top \in \mathbb{R}^{m \times n}$, $h : \mathbb{R}^m \to \mathbb{R}$ is nonsmooth, nonconvex, and separable, so $h(Ax) = \sum_{i=1}^m h_i(\langle a_i, x \rangle)$; while $g : \mathbb{R}^n \to \mathbb{R}$ is convex. We also consider *trimmed* extensions to robustify such models:

$$\min_{x,v} \quad \textstyle\sum_{i=1}^m v_i h_i(\langle a_i, x \rangle) + g(x), \quad \text{s.t.} \;\; v \in \triangle_\tau, \tag{2.2}$$

where $\triangle_\tau := \{v : v \in [0,1]^m, \;\; \sum_{i=1}^m v_i = \tau\}$ is the so called capped simplex. The auxiliary variables v detect $m - \tau$ outliers amongst the m observations as the optimization proceeds.

2.1.1 Examples

We present motivating examples for (2.1) before reviewing the literature and explaining the contributions. Each example is explained fully in Section 2.6. Examples 1-3 are **not weakly convex**, that is, they cannot be convexified by adding a quadratic. Weak convexity is a key property for the convergence theory of competitive methods covered in Section 2.1.3; RS does not require weak convexity. All of these examples can be robustified against outliers using *trimming* (2.2); trimming formulations are discussed at the end of Section 2.1.3 and the TRS approach for (2.2) is developed in Section 2.4.

Example 1 (Sharp phase retrieval)**.** *Given a complex matrix $A \in \mathbb{C}^{m\times n}$, the phase retrieval problem attempts to recover the full complex signal x using only moduli b:*

$$\min_{x\in\mathbb{C}^n} \quad \sum_{i=1}^{m} ||\langle a_i, x\rangle| - b_i|. \tag{2.3}$$

Example 2 (Semi-Supervised Classification)**.** *Logistic regression is a common approach for binary classification; training requires labeled examples. We solve an extended approach that makes use of both labeled and unlabeled data:*

$$\min_x \frac{\lambda}{2}\|x\|^2 + \sum_{i=1}^{l} \log(1+\exp(-b_i \langle a_i, x\rangle)) + \tau \sum_{i=l+1}^{m} \log(1+\exp(-|\langle a_i, x\rangle|)), \tag{2.4}$$

where a_i are features, b_i labels for the first s examples, and remaining $(m-l)$ examples are not labeled. The idea is to separate unlabeled examples as clearly as possible, regardless of which class they fall into.

Example 3 (Stochastic Shortest Path)**.** *Given a weighted graph on n nodes, we look for a policy that minimizes expected cost of path to target by selecting between one of two actions at each node. Let $U^k \in \mathbb{R}^{n\times n}$ and $v^k \in \mathbb{R}^n$ be the connectivity graphs and average node costs for $k = 1, 2$. Using the Bellman equation, the problem is formulated as*

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n} \left|\min\left\{\left\langle u_i^1, x\right\rangle + v_i^1 - x_i, \left\langle u_i^2, x\right\rangle + v_i^2 - x_i\right\}\right|, \tag{2.5}$$

where u_i^k is the i-th row vector of U^k and x_i is the best expected cost starting from node i.

Example 4 (Convex and Nonconvex Clustering)**.** *While K-means is the most widely used clustering method, an alternative is to solve the problem*

$$\begin{aligned} \min_X \quad & \frac{1}{2}\sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} R([DX]_{ij}) \\ s.t. \quad & [DX]_{ij} = x_i - x_j \end{aligned} \tag{2.6}$$

Table 2.1: Mapping motivating applications into class (2.1)

Example	$h(z)$	Linear map	$g(x)$
Phase retrieval	$\|\|z\| - b\|$	A	0
SS-LR	$\log(1 + \exp(-\|z\|))$	A	$\frac{\lambda}{2}\|x\|^2$
Stoch. path	$\|\min\{z - a, z - b\}\|$	U^1, U^2	0
Clustering	$R(z)$	D	$\frac{1}{2}\sum_{i=1}^{m} \|x_i - u_i\|^2$

where u_i is the reference data points and $X = [x_1, \ldots, x_m]$ are the decision variables, with R a regularization functional that acts to 'fuse' columns X into cluster representatives, and λ a regularization parameter that effectively controls the number of clusters. Classic approaches use a convex R, but we find a nonconvex R has significant advantages.

Table 2.1 maps Examples 1-4 to the templated objective (2.1). While the only theoretical requirement for $g(x)$ is convexity, in practice we assign simple smooth terms to g, so that we can implement fast subproblem solves. We can always take $g(x) = 0$ if necessary, rewriting a problem with multiple terms into a simple composition $h(Ax)$:

$$f_1(Bx) + f_2(x) = h\left(\begin{bmatrix} B \\ I \end{bmatrix} x\right), \quad \text{with} \quad A = \begin{bmatrix} B \\ I \end{bmatrix}, \text{ and } h(z_1, z_2) = f_1(z_1) + f_2(z_2).$$

The choice $g(x) = 0$ is allowed by the theory and common in practice.

2.1.2 RS for Nonconvex Composite Models

The core innovation of this work is to relax (2.1) and (2.2) by introducing an auxiliary variable w, and then use partial minimization over the original variables to develop efficient algorithms. In particular, we take the following 'relaxed' version of (2.1):

$$\min_{w,x} \; f_\nu(x, w) := h(w) + \frac{1}{2\nu}\|Ax - w\|^2 + g(x), \tag{2.7}$$

where w approximates Ax, decoupling the linear map from the nonsmooth, nonconvex h. The structure of (2.7) allows a partial minimization scheme. Define

$$g_\nu(w) := \min_{x} \; \frac{1}{2\nu}\|Ax - w\|^2 + g(x). \tag{2.8}$$

Problem (2.7) is now equivalent to

$$\min_{w} \; p_\nu(w) := h(w) + g_\nu(w). \tag{2.9}$$

Several observations can be made.

- Since g is convex, (2.8) can be solved efficiently, especially when g is also smooth.
- Conditioning of (2.9) is independent of A (see Table 2.2).
- The prox operator of h is easy to apply whenever h is separable.

These points affect the theoretical convergence and practical implementation of RS, and are made precisely in the analysis detailed in Section 2.3.

Contributions Our contributions are as follows.

- We develop relaxed models for (2.1) and (2.2), which are simple to optimize and very effective across a diverse set of applications (measured by application-specific metrics).
- We derive provably convergent algorithms for these relaxations, obtaining rates under different conditions on g and h. In contrast to recent work for nonsmooth nonconvex optimization, we do not assume that h is *weakly convex*. The new methods thus apply to a broader range of problems than prior art, and can handle e.g. exact phase retrieval and semi-supervised learning.
- We apply the approach to get promising application-specific results:
  - Exact phase retrieval, along with a trimmed robust extension;
  - Semi-supervised classification;
  - New direct approach for the stochastic shortest path problem;
  - A new scalable approach for convex and nonconvex clustering.

2.1.3 Related Work

Well-known approaches for nonsmooth, nonconvex problems include nonsmooth BFGS [223], Gradient Sampling [68], and derivative free methods (DFO), see e.g. [99]. These methods can be applied to problems more general than those in class (2.1) and (2.2); but they assume nothing about problem structure, and so there is little chance of scaling them to the semi-supervised SVMs and phase retrieval problems in our numerical examples, which have millions of variables. The lack of structure also limits the available convergence analysis: theoretical grounding for nonsmooth BFGS appears elusive; GS finds Clarke stationary points with unknown speed, while rates for DFO are known and must scale linearly with dimension.

More closely related to this paper is *convex-composite* optimization, which captures problems in classic nonlinear programming and more recently in large-scale machine learning. The

convex-composite class, see e.g. [66, 65]) generalizes both smooth and convex functions and is given by

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{m} h_i(c_i(x)) + g(x), \tag{2.10}$$

where g is a closed convex function, h_i are convex and Lipschitz, and c_i are smooth maps. The functions g and h_i provide an inference structure, while the maps c_i encode the data generating mechanism. Examples include exact penalty formulations of nonlinear programs [262, Section 17.2], robust phase retrieval (squared variant) [125], and matrix factorization [161]. Convex-composite problems have been extensively studied over the years [83, 281, 69, 372, 359, 149, 282], and have seen significant recent interest [224, 120, 83, 259, 121, 126].

The problem classes (2.1) and (2.2) fall outside of the convex-composite class any time h is both nonsmooth and nonconvex [1]. On the other hand, the nonconvex-composite class assumes that the data generating mechanism Ax is linear. An analysis of the natural superclass that allows nonsmooth nonconvex h and nonlinear maps c is left to future work.

Smoothing techniques are closely related to our approach; Moreau-Yosida smoothing (see Section 2.2) and related method of [258] are at the core of many well-known algorithms, including those of [38], [368], and [362]. If we partially minimize (2.7) with respect to w rather than with respect to x, we arrive at the problem

$$\min_x h_\nu(Ax) + g(x), \tag{2.11}$$

with h_ν analogous to the smoother discussed in [258]. However, since h is nonconvex, the function h_ν may also be nonsmooth and nonconvex (see the right panel of Figure 2.1), and (2.11) may be just as difficult to solve as the original problem. Minimizing over x instead leads to analyzable algorithms in the nonconvex-composite setting.

Another line of recent work combines stochastic gradient techniques with nonsmooth optimization [12, 107]. These approaches typically require stronger assumptions, such as smoothness or weak convexity of h. A function h is ρ-weakly convex when $h(\cdot) + \frac{\rho}{2}\|\cdot\|^2$ is convex. No function with 'inward kinks' can be weakly convex, which eliminates every one of our motivating examples.

Finally, we discuss the prior literature on trimmed estimation. Trimmed M-estimators were initially introduced by [301] in the context of least-squares regression. Recent work developed statistical theory [5, 363, 364] for robust high-dimensional applications, including lasso, graphical lasso, and sparse logistic regression. The Proximal Alternating Linearized Minimization (PALM) method of [52] can be used to find trimmed estimators (2.2) so long as the h functions are smooth and have Lipschitz continuous gradients. Better rates under the same assumptions are achieved by the algorithm of [12], who study the general formulation

$$\min_{x,v} \quad \sum_{i=1}^{m} v_i h_i(x) + g(x), \quad \text{s.t.} \;\; v \in \triangle_\tau, \tag{2.12}$$

[1] When h is smooth, $h(Ax)$ is smooth also and hence trivially convex-composite.

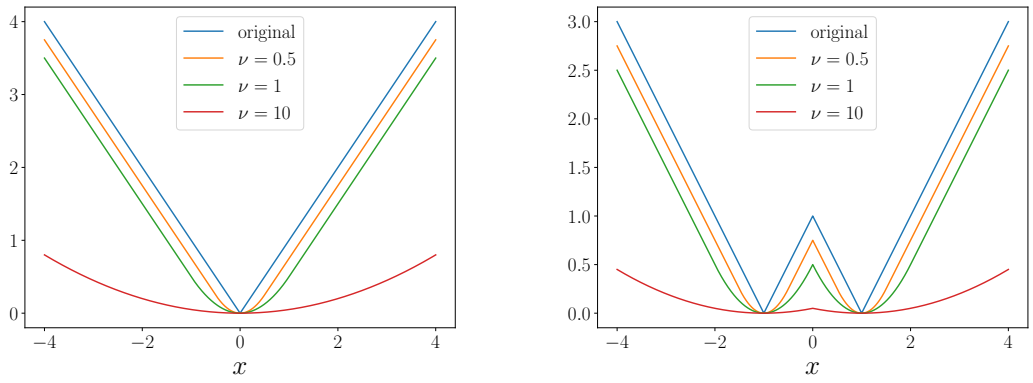


Figure 2.1: Moreau-Yosida smoothing for convex and nonconvex functions. The left figure plots smoothers for the convex function $h(x) = |x|$, while the right figure plots smoothers for the function $h(x) = ||x| - 1|$, which is not even weakly convex.

where $\tau < m$ is the estimated number of inliers, and the model $h : \mathbb{R}^m \to \mathbb{R}$ is smooth, while $g(x)$ is prox-bounded. Variables v separate inliers from outliers by finding elements $h_i(x)$ that disagree with the consensus, even as the consensus evolves due to updates of x. The set $\triangle_\tau$, called the *capped simplex*, as the intersection of the τ-simplex with the unit box, see (2.2). We extend the RS method to the nonconvex-composite class (2.2), so that we can trim nonsmooth nonconvex terms. This extension, called trimmed RS (TRS), allows for outlier detection and removal for any of the motivating examples, and we illustrate the power of the approach on the phase retrieval application in Section 2.6.1.

2.1.4 Road map

The paper proceeds as follows. RS is developed and analyzed in Section 2.3. The trimming extension and TRS are presented in Section 2.4. Practical considerations, including implementation, approximation and refinement, and discussed in Section 2.5, along with a comparison to the frequently used Alternating Directions Method of Multipliers (ADMM) problem in the convex setting. Detailed descriptions and results for the motivating applications are presented in Section 2.6. Proofs and technical details are collected in Appendix 2.8.

2.2 Notation and Preliminaries

In this section, we recall some basic notation that we will use throughout the manuscript. We will follow closely the monographs of [254] [295].

Euclidean Space. Throughout, we consider a Euclidean space, denoted by $\mathbb{R}^n$, with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|$. Given a linear map $A \colon \mathbb{R}^n \to \mathbb{R}^m$, the adjoint

$A^\top : \mathbb{R}^m \to \mathbb{R}^n$ is the unique linear map satisfying

$$\langle Ax, y\rangle = \langle x, A^\top y\rangle \qquad \text{for all } x \in \mathbb{R}^n, y \in \mathbb{R}^m.$$

The operator norm of A, defined as $\|A\| := \max_{\|u\|\le 1} \|Au\|$, coincides with the maximal singular value of A and satisfies $\|A\| = \|A^\top\|$.

Functions and Geometry. The extended-real-line is the set $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. The *domain* and the *epigraph* of any function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ are the sets

$$\operatorname{dom} f := \{x \in \mathbb{R}^d : f(x) < +\infty\}, \qquad \operatorname{epi} f := \{(x, r) \in \mathbb{R}^d \times \mathbb{R} : f(x) \le r\}.$$

We say that f is *closed* if its epigraph, $\operatorname{epi} f$, is a closed set. We assume that all functions that we encounter are *proper*, meaning they have nonempty domains and never take on the value $-\infty$. All the functions we consider in this paper are closed and proper.

Lipschitz Continuity. For any map $F : \mathbb{R}^n \to \mathbb{R}^m$, we set,

$$\operatorname{lip}(F) := \sup_{x \ne y} \frac{\|F(y) - F(x)\|}{\|y - x\|}.$$

In particular, we say that F is L-Lipschitz continuous, for some $L \ge 0$, if the inequality $\operatorname{lip}(F) \le L$ holds.

Fréchet and Limiting Subdifferentials. Consider 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

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

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 $\bar{x}$. We define the *limiting subdifferential* of f at $\bar{x}$, denoted $\partial f(\bar{x})$, to comprise all vectors v for which there exist sequences x_i and v_i, with $v_i \in \partial f(x_i)$ and $(x_i, f(x_i), v_i) \to (\bar{x}, f(\bar{x}), v)$.

Moreau Envelope and Proximal Mapping. For any function f and real $\nu > 0$, the *Moreau envelope* and the *proximal mapping* are defined by

$$f_\nu(x) := \inf_z \left\{ f(z) + \frac{1}{2\nu}\|z - x\|^2 \right\}, \tag{2.13}$$

$$\operatorname{prox}_{\nu f}(x) := \operatorname*{argmin}_z \left\{ f(z) + \frac{1}{2\nu}\|z - x\|^2 \right\}. \tag{2.14}$$

2.3 Convergence Analysis for RS

In this section, we develop and analyze a simple algorithm to find stationary points of the relaxed objective (2.9).

2.3.1 Proximal Gradient Method for the Relaxed Objective

Proximal gradient descent method (PGD) is a simple and powerful algorithm in the nonsmooth setting. It requires the objective to be a sum of smooth and 'prox-friendly' terms. Problem (2.9) is naturally viewed this way, since

- g_ν is smooth and its gradient is Lipschitz continuous, and
- h is prox-friendly; in particular it is separable.

Theorem 1. *Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a proper closed convex function that is bounded below, and $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map. Define function $g_\nu\colon \mathbb{R}^m \to \mathbb{R}$ and solution set x_ν to be,*

$$g_\nu(w) = \min_x \; g(x) + \frac{1}{2\nu}\|Ax - w\|^2,$$
$$x_\nu(w) = \operatorname*{argmin}_x \; g(x) + \frac{1}{2\nu}\|Ax - w\|^2.$$

For all $x_1, x_2 \in x_\nu(w)$, $x_1 - x_2 \in \mathrm{Null}(A)$. Moreover, g_ν is convex and C^1-smooth, with

$$\nabla g_\nu(w) = \tfrac{1}{\nu}(w - Ax), \quad \forall x \in x_\nu(w) \quad \textit{and} \quad \mathrm{lip}(\nabla g_\nu) \le \tfrac{1}{\nu}.$$

Proof. The proof is given in Appendix 2.8. □

Theorem 1 establishes the smoothness of g_ν. By the separability of h, $\mathrm{prox}_{\gamma h}$ decouples into a set of scalar optimization problems

$$\begin{aligned}\mathrm{prox}_{\gamma h}(v) &= \operatorname*{argmin}_w \tfrac{1}{2\gamma}\|w - v\|^2 + h(w) \\ &= \begin{bmatrix} \operatorname{argmin}_{w_1} \frac{1}{2\gamma}(w_1 - v_1)^2 + h_1(w_1) \\ \vdots \\ \operatorname{argmin}_{w_m} \frac{1}{2\gamma}(w_m - v_m)^2 + h_m(w_m) \end{bmatrix}.\end{aligned}$$

Even though h is nonconvex and nonsmooth, scalar problems are typically easy to solve. To implement the motivating examples, we found closed form solutions for the prox operators in examples 1, 3, 4, and implemented a Newton method for semi-supervised logistic regression in example 2. Some h require root-finding or bi-section techniques, but due to the separability assumption, these methods need only be applied to scalar problems.

The PGD algorithm is detailed in Algorithm 1.

Algorithm 1 Proximal Gradient Descent for $h(w) + g_\nu(w)$

Input: w^0

Initialize: $k = 0$

while not converge **do**

  $w^{k+1} \leftarrow \text{prox}_{\nu h}(w^k - \nu \nabla g_\nu(w^k))$

  $k \leftarrow k + 1$

end while

Output: w^k

We can write the w-update in Algorithm 1 explicitly:

$$\text{prox}_{\nu h}(w^k - \nu \nabla g_\nu(w^k)) = \text{prox}_{\nu h}(Ax^k), \quad x^k(w^k) \in \underset{x}{\text{argmin}} \; g(x) + \frac{1}{2\nu}\|Ax - w^k\|^2. \quad (2.15)$$

In the next section, we analyze the behavior of Algorithm 1 under different assumptions.

2.3.2 Convergence Analysis

The goal for Algorithm 1 is to find the stationary point for (2.9), defined as follows.

Definition 1 (Stationary Point)**.** *A point $\bar{w} \in \mathbb{R}^m$ is called a stationary point for* (2.9) *if*

$$0 \in \nabla g_\nu(w) + \partial h(w).$$

Equivalently, we can write

$$0 \in \left\{ \partial h(\bar{w}) + \tfrac{1}{\nu} \left(I - A \left(\partial g + \frac{1}{\nu} A^\top A \right)^{-1} A^\top \right) \bar{w} \right\} := \mathcal{S}(\bar{w}).$$

where $\left(\partial g + \frac{1}{\nu} A^\top A\right)^{-1} \bar{w}$ is a nonlinear (possibly multi-valued) operator that gives the set of solutions $x(\bar{w})$ to the problem in (2.15)*.*

Motivated by this definition, we define the following quantity to measure optimality.

Definition 2 (Optimality Condition)**.** *We denote*

$$T_\nu(w) = \min \left\{ \|v\|^2 : v \in \mathcal{S}(\bar{w}) \right\}, \quad (2.16)$$

as the optimality condition of (2.9)*.*

Convergence rates of Algorithm 1 depends on additional assumptions on h and g, and are summarized in Table 2.2. All proofs for this section are collected in Appendix 2.8.

We now analyze Algorithm 1 under different assumptions on h and g. We start the analysis under the weakest assumptions (h prox-bounded and g closed convex), and continue

	Rate of Convergence
Assumption 1	$\bar{T}_\nu^k \leq \frac{2}{\nu k}[p_\nu(w^0) - p_\nu^*]$
Assumption 2	$p_\nu(w^k) - p_\nu^* \leq \frac{\|w^0 - w^*\|^2}{2\nu(k+1)}$
Assumption 3	$\|w^{k+1} - w^*\|^2 \leq \frac{1}{1+\alpha\nu}\|w^k - w^*\|^2$
Assumption 4	$\|w^{k+1} - w^*\| \leq \frac{1}{\alpha\nu}\|w^k - w^*\|^2$

Table 2.2: Summary of convergence rates for Algorithm 1. We denote $\bar{T}_\nu^k$ as the average of quantity (2.16) in k steps, namely $\frac{1}{k}\sum_{i=1}^k T_\nu(x^i, w^i)$. p_ν^* and w^* are the optimal objective value and optimal solution in the convex case.

to much stronger assumptions (h has a sharp minimum and $g = 0$). The latter results help us understand the empirically observed local behavior of Algorithm 1.

In order for problem (2.9) to be well-defined, we assume that p_ν is bounded below, and that the minimum can be attained, and define

$$p_\nu^* = \min_w \ p_\nu(w), \quad w^* = \operatorname*{argmin}_w \ p_\nu(w).$$

General Case

Assumption 1. *h is prox-bounded, so that there exists a $\overline{\nu}$ with $\operatorname{prox}_{\nu h}(x)$ nonempty for all x and $\nu > \overline{\nu}$; g is convex.*

Theorem 2. *If Assumption 1 holds, the iterates generated by Algorithm 1 satisfy,*

$$\tfrac{1}{\nu}A(x^{k-1} - x^k) \in \partial h(w^k) + \tfrac{1}{\nu}(w^k - Ax^k), \quad \textit{where} \quad 0 \in \partial g(x^k) + \tfrac{1}{\nu}A^\top(Ax^k - w^k).$$

moreover,

$$\bar{T}_\nu^k := \frac{1}{k}\sum_{i=1}^k T_\nu(w^i) \leq \frac{1}{k}\sum_{i=1}^k \left\|\tfrac{1}{\nu}A(x^{i-1} - x^i)\right\|^2 \leq \frac{2}{\nu k}[p_\nu(w^0) - p_\nu^*].$$

We thus obtain a sublinear rate of convergence for the optimality condition. Note that this rate is independent of linear map A.

Convex Case

Assumption 2. *h and g are both proper closed convex functions.*

In this case, $h(w) + g_\nu(w)$ is a sum of a convex nonsmooth and convex smooth functions. This problem class has been exhaustively studied; see e.g. the survey of [275]. The FISTA algorithm [35], detailed in Algorithm 2, can achieve faster convergence rates for this problem than Algorithm 1.

Theorem 3. *If Assumption 2 holds, the iterates generated by Algorithm 1 satisfy,*

$$p(w^k) - p_\nu^* \leq \frac{\|w^0 - w^*\|^2}{2\nu(k+1)}.$$

Algorithm 2 FISTA for $h(w) + g_\nu(w)$

Input: w^0

Initialize: $k = 0$, $a_0 = 1$, $v^0 = w^0$

1: **while** not converge **do**

2: $w^{k+1} \leftarrow \text{prox}_{\nu h}\left(v^k - \nu \nabla g_\nu(v^k)\right)$

3: $a^{k+1} \leftarrow \frac{1+\sqrt{1+4(a^k)^2}}{2}$

4: $v^{k+1} \leftarrow w^{k+1} + \frac{a^{k-1}}{a^{k+1}}(w^{k+1} - w^k)$

5: $k \leftarrow k + 1$

6: **end while**

Output: w^k

Theorem 4. *If Assumption 2 holds, the iterates generated by Algorithm 2 satisfy [35]:*

$$p_\nu(w^k) - p_\nu^* \leq \frac{2\|w^0 - w^*\|^2}{\nu(k+1)^2}.$$

Strongly Convex Case

In two of our motivating examples, we take $g = 0$. In this case, we have a closed form solution for (2.8),

$$g_\nu(w) = \frac{1}{2\nu}\|(I - P_A)w\|^2, \quad \text{where} \quad P_A = A(A^\top A)^\dagger A^\top,$$

and $\dagger$ denotes the pseudo inverse.

Assumption 3. *h is α-strongly convex and $g = 0$.*

Theorem 5. *When Assumption 3 holds, the iterates generated by Algorithm 1 satisfy,*

$$\|w^{k+1} - w^*\|^2 \leq \frac{1}{1 + \alpha\nu}\|w^k - w^*\|^2.$$

That is, we obtain a linear convergence rate in this case.

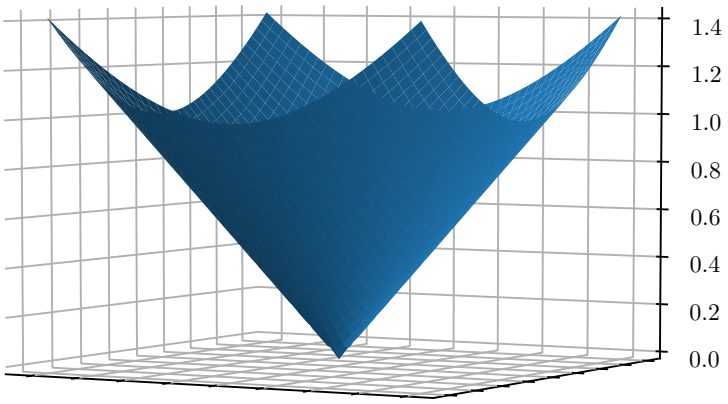


Figure 2.2: Local function values grow quickly away from a *sharp minimum*.

Sharp Minima Case

The final assumption concerns *sharp minima*, see [4, 101, 187, 280, 70] and Figure 2.2.

Definition 3. *We say the minimizer w^* of p_ν is a sharp minimum, if there exist $\delta, \alpha > 0$, such that,*

$$p_\nu(w) - p_\nu(w^*) \geq \alpha\|w - w^*\|, \quad \forall w \in \{w : \|w - w^*\| \leq \delta\}.$$

Assumption 4. *h is proper closed convex, $g = 0$ and w^* is a sharp minimum of p_ν.*

Theorem 6. *If Assumption 4 holds, and there exists an iteration K with that,*

$$\|w^k - w^*\| \leq \delta$$

then for all $k \geq K$, iterates generated by Algorithm 1 satisfy

$$\|w^{k+1} - w^*\| \leq \min\left\{\|w^k - w^*\|, \frac{1}{\alpha\nu}\|w^k - w^*\|^2\right\}.$$

A sharp minimum gives us a local quadratic convergence rate.

2.4 Trimmed Nonconvex-Composite Models

We apply an analogous relaxation technique to problem class (2.2), obtaining the extended problem

$$\min_{v,x,w} \; f_\nu^t(x, w, v) := \sum_{i=1}^{m} v_i h_i(w_i) + g(x) + \frac{1}{2\nu}\|Ax - w\|^2, \quad \text{s.t. } v \in \triangle_\tau, \tag{2.17}$$

Algorithm 3 Block-Coordinate Descent for (2.17)

Input: w^0, v^0, γ

Initialize: $k = 0$

1: **while** not converged **do**

2: $w^{k+1} \leftarrow \text{prox}_{\nu(\langle v, H\rangle)}\left(w^k - \nu \nabla g_\nu(w^k)\right)$

3: $v^{k+1} \leftarrow \text{proj}_{\triangle_\tau}(v^k - \gamma H(w^{k+1}))$

4: $k \leftarrow k + 1$

5: **end while**

Output: w^k

where each function h_i is nonsmooth and nonconvex.

We use the notation $H(w) = [h_1(w_1), \ldots, h_m(w_m)]^\top$, so that $\sum_{i=1}^m v_i h_i(w_i) = \langle v, H(w)\rangle$.

Just as in Section 2.3, we partially minimize in x, reducing (2.17) to problem

$$\min_{v,w} \; p_\nu^t(w, v) := \sum_{i=1}^m v_i h_i(w_i) + g_\nu(w), \quad \text{s.t. } v \in \triangle_\tau \tag{2.18}$$

The structure of (2.17) suggests a coordinate-descent algorithm detailed in Algorithm 3.

The operator $\text{prox}_{\nu(\langle v,H\rangle)}$ decouples across coordinates; for each nonzero v_i, we have

$$\text{prox}_{\nu(\langle v,H\rangle)}(\bar{w}) = \begin{bmatrix} \text{argmin}_{w_1} \frac{1}{2v_1\nu}(w_1 - \bar{w}_1)^2 + h_1(w_1) \\ \vdots \\ \text{argmin}_{w_m} \frac{1}{2v_m\nu}(w_m - \bar{w}_m)^2 + h_m(w_m) \end{bmatrix}.$$

We now develop a convergence analysis for Algorithm 3. Our goal is to find the stationary point of (2.18), defined as follows.

Definition 4. *We call the pair $(\bar{w}, \bar{v})$ a stationary point of* (2.18) *when*

$$0 \in \begin{bmatrix} \bar{v}_1 \partial h_1(\bar{w}_1) \\ \vdots \\ \bar{v}_m \partial h_m(\bar{w}_m) \end{bmatrix} + \nabla g_\nu(\bar{w}) := \mathcal{S}_w^t(\bar{w}, \bar{v}), \quad 0 \in H(\bar{w}) + \partial\delta(\bar{v}|\triangle_\tau) := \mathcal{S}_v^t(\bar{w}, \bar{v}).$$

We define the following quantity to characterize stationarity:

$$T_\nu^t(w, v) = \min\left\{\tfrac{\nu}{2}\|s\|^2 + \alpha\|r\|^2 : s \in \mathcal{S}_w^t(w, v),\, r \in \mathcal{S}_v^t(w, v)\right\}.$$

The convergence result is detailed in Theorem 7.

Theorem 7. *Denote by w^k and v^k the iterates generated by Algorithm 3. We have the following inequality,*

$$T_\nu^t(w^{k+1}, v^{k+1}) \le p_\nu^t(w^k, v^k) - p_\nu^t(w^{k+1}, v^{k+1}).$$

Moreover, by manipulating this inequality we obtain

$$\frac{1}{k}\sum_{i=1}^{k} T_\nu^t(w^i, v^i) \leq \frac{1}{k}[p_\nu^t(w^0, v^0) - p_\nu^t(w^k, v^k)],$$

which gives a sublinear rate of convergence for Algorithm 3.

Proof. The proof is given in Appendix 2.8. □

2.5 Numerical Comparisons, Continuation, and Inexact Strategies

In this section we provide numerical experiments that help to better understand Algorithm 1. In Section 2.5.1, we compare with the Alternating Directions Method of Multipliers (ADMM) in the convex setting. The iterations of ADMM are similar to those of Algorithm 1, with the augmented Lagrangian parameter ρ in ADMM analogous to the relaxation parameter $\frac{1}{\nu}$ for RS. However, ADMM performs worse than RS in a direct comparison: it needs a larger number of iterations to achieve a specified error tolerance across choices of ρ and ν, and RS can achieve better practical performance, depending on the application. In Section 2.5.2, we discuss continuation strategies in ν, that become important when RS is used iteratively to approximate the original problem (2.1). Finally, in Section 2.5.3 we consider large-scale problems where problem (2.8) cannot be solved in closed form, and iterative methods are required.

2.5.1 Comparison to ADMM in the Convex Setting

Although Algorithm 1 bears a strong resemblance to the ADMM algorithm (Algorithm 4, see e.g. [55]), they are fundamentally different:

- ADMM is a primal-dual method solving (2.1) while Algorithm 1 is a primal-only approach for solving the relaxation (2.7).

- ADMM has convergence guarantees for convex objectives[2], while Algorithm 1 is provably convergent both convex and nonconvex optimization problems.

We compare the two algorithms on a simple objective.

Example 5. *Consider ℓ_1 linear regression,*

$$\min_x \|Ax - b\|_1. \tag{2.19}$$

[2]Convergence for nonconvex problems requires additional assumptions, see e.g. [352]

Algorithm 4 ADMM for convex $h(Ax) + g(x)$

Input: x^0, ρ, α

Initialize: $k = 0$, w^0, u^0

while not converge **do**

$x^{k+1} \leftarrow \operatorname{argmin}_x g(x) + \langle u^k, Ax - w^k \rangle + \frac{\rho}{2}\|Ax - w^k\|^2$

$w^{k+1} \leftarrow \operatorname{prox}_{h/\rho}(Ax^{k+1} - u^k/\rho)$

$u^{k+1} \leftarrow u^k - \alpha(Ax^{k+1} - w^{k+1})$

$k \leftarrow k + 1$

end while

Output: w^k

The quadratic relaxation (2.1) *is given by*

$$\min_{x,w} \|w - b\|_1 + \frac{1}{2\nu}\|Ax - w\|^2. \tag{2.20}$$

Here $A \in \mathbb{R}^{m\times n}$ *and* $x_t \in \mathbb{R}^n$ *are generated from standard Gaussian distribution, and* $b = Ax_t + \epsilon + o$ *with* ϵ *to be random Gaussian noise, and* o *to be sparse outliers. We denote the solution to* (2.19) *as* x_{ℓ_1} *and the solution to* (2.20) *as* x_ν.

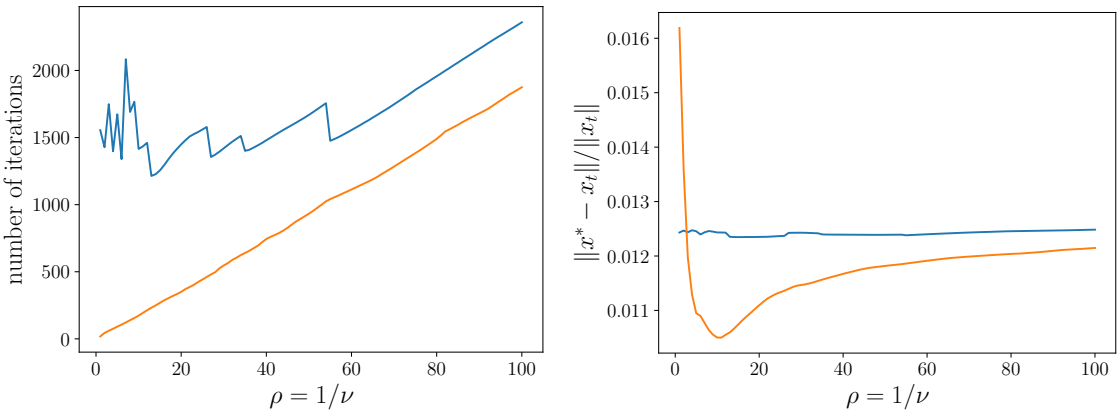


Figure 2.3: Comparison between Algorithm 1 and ADMM. Left: number of iterations required by ADMM (blue) and Algorithm 1 (orange) to converge to a fixed tolerance, as a function of varying $\rho = 1/\nu$. Right: relative error of the solution obtained from ADMM (blue) and Algorithm 1 (orange) with respect to x_t.

The numerical results are shown in Figure 2.3. In the experiments, we fix the augmented Lagrangian coefficient ρ in ADMM to be equal to $1/\nu$, and this quantity from 1 to 100. We

then plot the number of iterations required to hit a specified error tolerance, as well as the relative error of the recovered solution with respect to x_t.

As shown in the left plot of Figure 2.3, the number of iterations of Algorithm 1 grows linearly as a function of $1/\nu$, but is always below the number required by ADMM. The right figure of Figure 2.3 tells an interesting story. The relaxation may be *more accurate* than the original problem, depending on the application. When $\rho = 1/\nu = 10$, the solution of the relaxed formulation (2.20) is closer to the true model that that of (2.19), and Algorithm 1 can solve (2.20) much faster than ADMM can solve (2.19). Both the improvement in accuracy and the computational advantage persist as $\nu \downarrow 0$. In this problem, ADMM and RS iterations have exactly the same complexity, so the iterations comparison tells the full story.

2.5.2 Continuation

In the previous section, the solution obtained from the relaxed objective was closer to the true model. In other cases, such as noiseless phase retrieval, (2.7) and (2.1) can share a minimizer at a large value of ν. However, more generally we may want to use (2.7) as an approximation to (2.1), in which case we want to explore continuation schemes with $\nu \downarrow 0$.

Theorem 8. *If h is L-Lipschitz continuous and $(\bar{x}, \bar{w})$ is a stationary point of (2.7), we have,*

$$\|A\bar{x} - \bar{w}\| \leq L\nu.$$

Moreover, when $A\bar{x} = \bar{w}$, we know $\bar{x}$ is also a stationary point of (2.1).

From Theorem 8 we know that, as ν goes to 0, solutions of (2.7) approach the solution set of (2.1). This yields a simple continuation strategy. Using the setting of Example 5, we take a decreasing positive sequence $\{\nu^k\}$, and initialize x_{ν^k} at the previous solution $x_{\nu^{k-1}}$.

We generate A at different dimensions $m \in \{500, 1000, 2000, 5000, 10000\}$ with $n = 200$, and compare the results from the continuation strategy of Algorithm 1 continuation with the Julia Convex Package (which uses the splitting cone solver (SCS)). We check the final objective for (2.19), as well as the run times. Results are shown at Figure 2.4.

Algorithm 1 gets a slightly lower objective value than the SCS algorithm; it is also far faster in terms of run-time, as shown in Figure 2.4. We emphasize that here SCS and Algorithm 1 are solving the same objective (2.19), since we drive $\nu \downarrow 0$ using a continuation strategy.

2.5.3 Inexact Solutions

Each iteration of Algorithm 1 requires solving a linear system. The potential drawback of Algorithm 1 is the computational cost for problem (2.8), especially for large scale problems. In many imaging applications, A is an orthogonal operator, like the Fourier transform, Wavelet transform or Hadamard matrix; as a result, problem (2.8) in Algorithm 1 is tractable at acale. In more general applications, when the matrix A is of moderate size, $A^\top A + \frac{1}{\nu} I$ can be

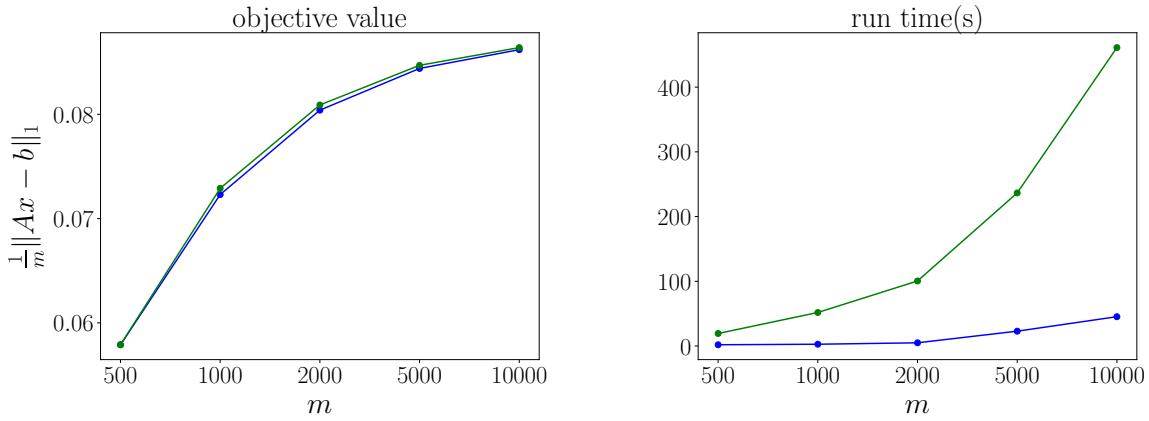


Figure 2.4: Comparison between Algorithm 1 continuation and the Julia Convex package with SCS. Left: objective values for (2.19) Algorithm 1 (blue) and SCS (green) as a function of m; the continuation approach finds the same or lower objective value as SCS. Right: run times for Algorithm 1 (blue) with SCS (green) as a function of m. The total work of the continuation approach is far less than required by SCS as m increases.

pre-factored, and the factors used to solve (2.8). However, for large-scale systems A may only be accessible through matrix-vector multiplication, and inexact solves of (2.8) are required to make Algorithm 1 practical.

Again using the setup in Example 5, we consider iterative methods, including preconditioned CG [186] and LSQR [274] to solve the problem for large n.

CondNum	Alg. 1 iters	Total BFGS	Alg. 1 time(s)
1	12	12	0.74
10	15	1099	18.28
20	20	1040	18.65
50	35	1054	22.87
100	60	1104	32.28

Table 2.3: Iterations and run times for Alg. 1 with BFGS solving (2.8). As the condition number grows, the total number of BFGS iterations used by Alg. 1 stays bounded.

In this experiment, we choose $m = 5000$, $d = 1000$, $\nu = 1$ and generate random matrices A with different condition numbers. We use BFGS (see e.g. [150]) as the inner solver for (2.8). As the condition number increases, Algorithm 1 behaves quite well in the large-scale setting, as the total number of inner iterations stays bounded.

2.6 Machine Learning Applications

In this section, we give more detailed explanations for the motivating examples, and present numerical experiments and results. Phase Retrieval and its trimmed variant is presented in Section 2.6.1. Semi-supervised classification is considered in Section 2.6.2. The stochastic shortest path problem is developed in Section 2.6.3. New approaches for convex and nonconvex clustering are discussed in Section 2.6.4.

2.6.1 Sharp Phase Retrieval

Phase retrieval was originally introduced in signal processing for the X-ray crystallography problem [182, 253] and arises in such diverse fields as microscopy [252, 151, 118], holography [147, 322], neutron radiography [6], optical design [143], adaptive optics, and astronomy. For a detailed review of applications and algorithms, see the survey of [236].

Many algorithms has been studied by [145, 146, 160]. Recently, phase retrieval has gained some attention with the work of [78, 127, 133] and [108].

We consider an exact formulation of phase retrieval problem,

$$\min_x \||Ax| - b\|_1 \tag{2.21}$$

where x is the signal we want to recover, $|\cdot|$ is the modulus of a complex number, and b are the observed moduli obtained from linear observations A of the true signal. We take $h_i(z) = ||z| - b_i|$, $g(x) = 0$ and optimize

$$\min_{x,w} \||w| - b\|_1 + \frac{1}{2\nu}\|Ax - w\|^2. \tag{2.22}$$

We assume there is no noise in the experiment, so that $b = |Ax^*|$. In this case, (2.22) and (2.21) share the same solution.

We test Algorithm 1 on a large scale phase retrieval problem. We use a color image[3] that is 2048×2048, with $m = 9 \times 2^{22}$ observations and $n = 3 \times 2^{22}$ unknowns. We define H_n to be a normalized Walsh-Hadamard transform:

$$H_n \in \{-1, 1\}^{n\times n}/\sqrt{n}, \quad H_n = H_n^\top, \quad H_n^2 = I.$$

The linear operator A is given by

$$A = \begin{bmatrix} H_n S_1 \\ \vdots \\ H_n S_k \end{bmatrix} \in \mathbb{R}^{kn\times n},$$

with $k = 4$ and $S_1, \ldots, S_k \in \mathrm{diag}(\{-1, 1\}^n)$.

[3] http://getwallpapers.com/wallpaper/full/8/5/0/651422.jpg

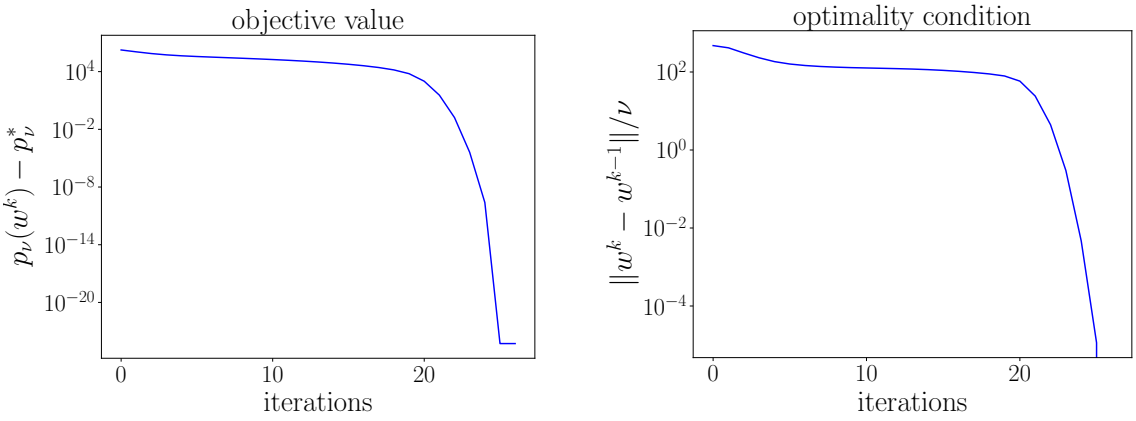


Figure 2.5: Convergence history for large-scale phase retrieval.

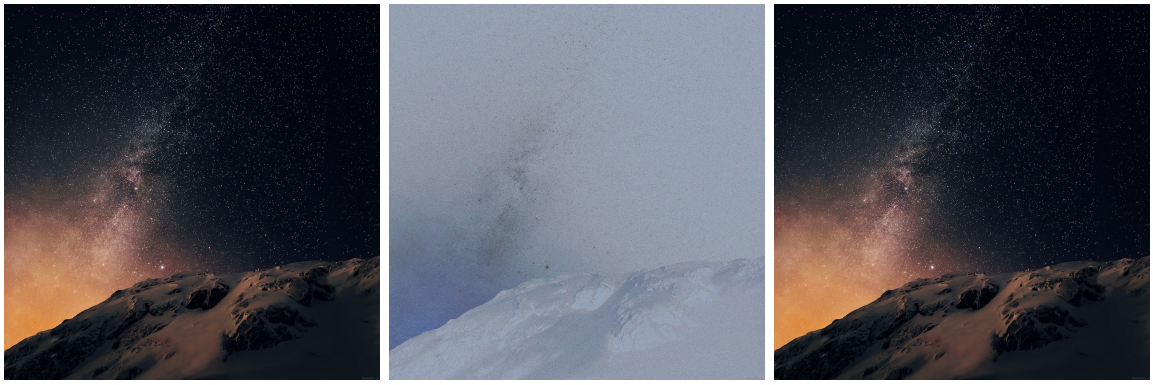


Figure 2.6: Large example ($d = 3 \times 2^{22}, m = 3 \times 2^{22}$). Original picture (left), initial point (middle), and final result (right).

The results are shown in Figures 2.5 and 2.6. The initialization algorithm works well, and Algorithm 1 converges within 30 iterations. Even though hypotheses of Theorem 6 do not hold (h is nonconvex), we expect a local quadratic rate of convergence since the minimum is sharp, and we observe this rate Figure 2.5.

Comparison to State-of-the-Art Phase Retrieval Algorithms. We compare Algorithm 1 with other methods developed by [127] and by [108]. We summarize the results in Table 2.4.

Algorithm 1 uses fewer matrix vector multiplications (fast Hadamard transforms) to obtain the solution, compared to recently developed phase retrieval algorithms. The counts include initialization, with Algorithm 1 using 10 power iterations to initialize, while [108]

	objective	picture size	dimension	# meas	# FHT
Algorithm 1	$\|\|\|Ax\| - b\|\|_1$	2048^2	$n = 3 \times 2^{22}$	$m = 3n$	518
[127]	$\|(Ax)^2 - b\|_1$	1024^2	$n = 3 \times 2^{20}$	$m = 3n$	15100
[108]	$\|(Ax)^2 - b\|_1$	2048^2	$n = 3 \times 2^{22}$	$m = 3n$	1530

Table 2.4: Comparison summary. FHT stands for fast Hadamard transfrom. The number of FHTs include those used during initialization.

start at random point. For this problem, Algorithm 1 is minimizing a different objective than the other methods, see Table 2.4. However, we can compare the Hadamard counts directly since all methods recover the true phase.

Trimmed Phase Retrieval. The measurements of the magnitude can be corrupted due to detector malfunction, heteroscedastic noise, or physical limitations. A robust extension of phase retrieval is needed in these situations. We use the trimmed extension of (2.21):

$$\min_{v,x} \sum_{i=1}^{m} v_i |\,|\langle a_i, x\rangle\,| - b_i|, \quad \text{s.t. } v \in \triangle_\tau, \tag{2.23}$$

where τ indicates the estimated number of good measurements. This is a nonsmooth trimming problem, and we use TRS, see Section 2.4. The relaxed trimmed phase retrieval objective is given by

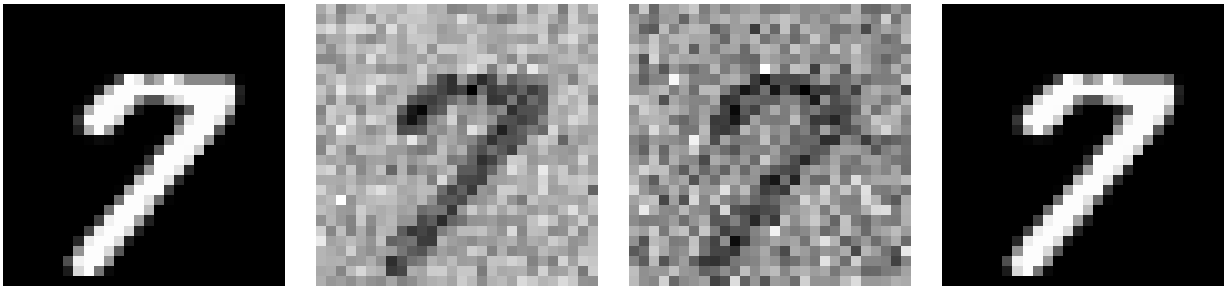
$$\min_{w,v,x} \ \frac{1}{2} \sum_{i=1}^{m} v_i ||w_i| - b_i|^2 + \frac{1}{2\nu}\|Ax - w\|^2, \quad \text{s.t. } v \in \triangle_\tau. \tag{2.24}$$

In the experiments, we use a small MNIST[4] picture as the data source with dimension $n = 28 \times 28 = 784$. We take $m = 5n$, measurements, and corrupt 30% of them by replacing the measurements with large scalar 1000. We then solve both (2.22) and (2.24). Trimming makes a significant difference in the quality of the recovered image, see Figure 2.7.

2.6.2 Semi-Supervised Classification

Classification is a fundamental problem in machine learning. Logistic Regression ([247]) and Support Vector Machines (SVMs, see [100]) are used widely for binary classification; training requires labeled examples. In many applications, labeling the data can a slow, costly and error-prone process. Semi-supervised learning attempts to use both labeled and unlabeled data to improve accuracy (relative to using only labeled data).

[4] `http://yann.lecun.com/exdb/mnist/`



(a) (b) (c) (d)

Figure 2.7: The advantages of trimming phase retrieval: (a) is the true data source, (b) is the initial starting point, (c) phase retrieval results using (2.22), (d) trimmed phase retrieval results using (2.24).

Logistic regression for binary classification is both easily formulated and widely used. We consider the semi-supervised logistic regression (SSLR).

Building on early work for semi-supervised classification in the pattern recognition community (see survey in [248]), [7] proposed a variant of SSLR, building a discriminant logistic model and using a Classification Expectation Maximization (CEM) algorithm to solve the resulting formulation. The work of [7] and follow-up papers (e.g. [238]) share a key theme: they estimate posterior probabilities of class labels, which are then used in the maximization step. The idea of taking expectations over class labels brings the Expectation-Maximization (EM) algorithm to bear on the model.

Our approach to semi-supervised logistic regression is inspired by transductive SVMs, introduced by [341]. The more modern variant of the problem is often called the semi-supervised SVM (S^3VM), see e.g. work of [87].

Following the intuition of transductive SVMs, we want to solve the logistic regression problem while separating unlabeled data as well as possible, regardless of the label. This leads to an intuitively simple nonsmooth, nonconvex problem

$$\min_x \sum_{i=1}^{l} \log(1+\exp(-b_i \langle a_i, x\rangle)) + \gamma \sum_{i=l+1}^{m} \log(1+\exp(-|\langle a_i, x\rangle|)) + \frac{\lambda}{2}\|x\|^2, \qquad (2.25)$$

where $a_i \in \mathbb{R}^n$ is the data image, $b_i \in \{-1, 1\}$ is the label and γ controls the weight of the semi supervise part. Without loss of generality, we assume only the first l images are labeled. Geometrically, when data is labeled, the direction to push the classifier is determined; when data is unlabeled, we tend to push the classifier in both ways depend on its current position.

Problem (2.25) is different from all previous SSLR formulations, and in particular does not require an EM algorithm; it can be optimized directly. Problem (2.25) falls squarely into

the framework we proposed in this paper, and the relaxed objective can be written as,

$$\min_{x,w} \sum_{i=1}^{l} \log(1+\exp(-b_i w_i)) + \gamma \sum_{i=l+1}^{m} \log(1+\exp(-|w_i|)) + \frac{1}{2\nu}\|Ax-w\|^2 + \frac{\lambda}{2}\|x\|^2. \quad (2.26)$$

If we treat (2.26) as a specification of (2.7) we have,

$$g(x) = \frac{\lambda}{2}\|x\|^2, \quad h_i(z) = \begin{cases} \log(1+\exp(-b_i z)), & i \le l \\ \log(1+\exp(-|z|)), & i > l \end{cases},$$

and when $i > l$ we know that h_i is nonconvex and nonsmooth.

To apply Algorithm 1, closed form solution of (2.8) can be obtained. We also need to calculate the proximal operator of h_i. For $i \le l$, the prox-subproblems is smooth and convex. For $i > l$, i.e. for the unlabeled examples, the prox problem in each coordinate requires solving the scalar problem,

$$\min_{w_i} \frac{1}{2\nu}(w_i - \overline{w}_i)^2 + \gamma \log(1+\exp(-|w_i|)).$$

The optimal z will necessarily have the same sign as $\overline{z}$, and so we can rewrite the problem

$$\min_{|w_i|} \frac{1}{2\nu}(|w_i| - |\overline{w}_i|))^2 + \gamma \log(1+\exp(-|w_i|)).$$

This is again a smooth and convex problem in w, so we can apply Newton's method to find $|\hat{w}_i|$. The solution $\hat{w}_i$ is then immediately obtained by $\hat{w}_i = |\hat{w}_i|\ \text{sign}(\overline{w}_i)$.

Our goal in the experimental results is to illustrate the simplicity and flexibility of the new SSLR concept. We leave a comprehensive comparison with prior art on semi-supervised classification to future work.

Figure 2.8 shows the convergence result for run of the algorithm, with parameters $m = 12665$, $l = 254$ (2% of data labeled), $\lambda = 0.1$, $\gamma = 0.1$ and $\nu = 1$. Consistently with Theorem 3, when h is nonconvex, Algorithm 1 has a sublinear rate.

To evaluate the results, we focus on prediction accuracy as a function of the γ parameter in (2.26), and fix $\lambda = 0.1$, $\nu = 1$. We let γ range among $0, 0.1, \ldots, 0.9, 1$. We use two sets of MNIST data, considering binary classification of digit pairs (0, 1) and (4, 9). For each choice of γ, we conduct 20 random trails and record the mean and variance of the test accuracy.

Testing errors are shown in Figure 2.9. Several observation can be made.

- (4, 9) yields a harder classification problem compared with (0, 1). For each ratio of labeled to unlabeled data, test accuracy for (4, 9) is lower than for (0, 1).

- Semi-supervised learning helps more for the MNIST dataset when we have very few labeled datapoints.

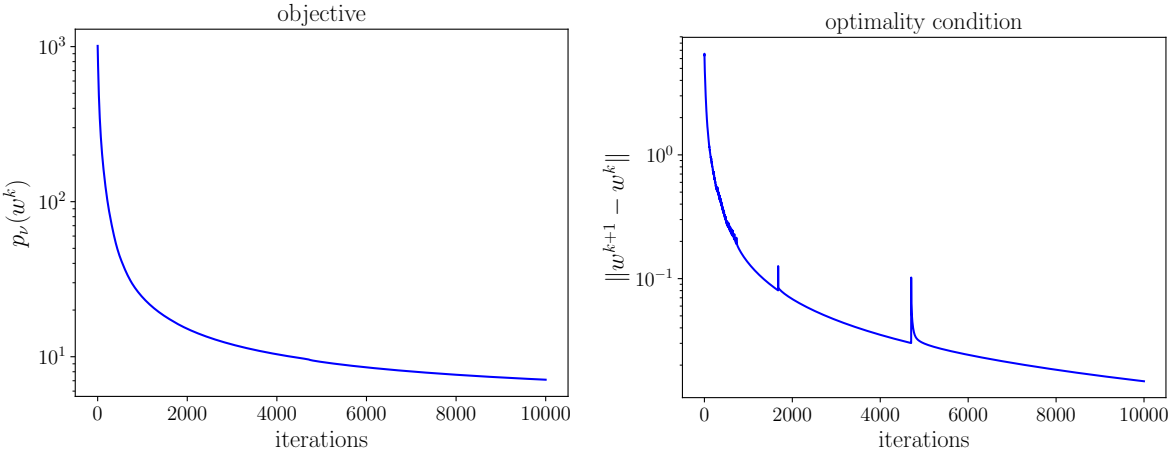


Figure 2.8: Convergence plot for semi-supervised Logistic Regression.

- The variance of accuracy results increases with γ (as we pay more attention to unlabeled data), and decrease with ratio of labeled to unlabeled data.

We see the lowest test error for $\gamma = 0.1$ across all experiments.

The results show that some degree of improvement is readily obtained from the SSLR strategy, and that the proposed approach can easily handle the new type of optimization problem. We leave extensions to more powerful learning models and comparisons with the robust literature on semi-supervised classification to future work.

2.6.3 Stochastic Shortest Path

In this experiment, we consider the stochastic shortest path problem described by [42]. For a review of the history of shortest path problem, please check [308]. As shown in Figure 2.11, the version we consider looks for the minimum expected cost path from from node A to node B, given a certain graph structure. At each node, we select between two graphs, then take a step by uniformly sampling available paths of the chosen graph to move to an adjacent node, paying the specified cost.

The specific example we consider contains $n = 25$ nodes. Two graphs are generated randomly, along with the cost matrices $C^1, C^2 \in \mathbb{R}^{n\times n}$ for each graph, with C^k_{ij} defined as the cost[5] to move from node i to node j within graph k. We also let $U^1, U^2 \in \mathbb{R}^{n\times n}$ denote the connectivity matrices, with entry U^k_{ij} encoding the probability that node i moves to node j within graph k.

[5]The cost matrices C^k are generated uniformly at random.

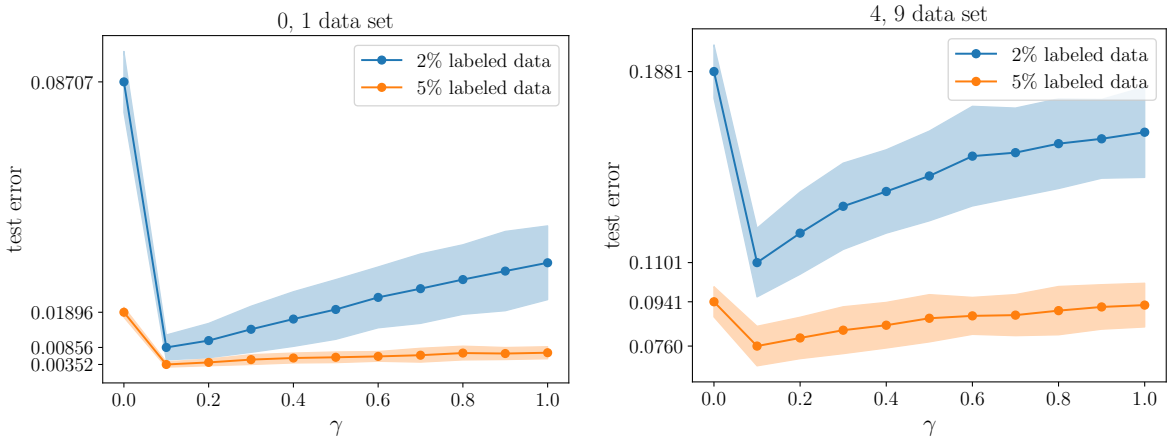


Figure 2.9: Testing errors of semi-supervised logistic regression. Left: results of the (0, 1) classification experiment. Right: results of the (4, 9) classification experiment. Both plots show the test errors as a function γ, with 2% labeled data (blue) and 5% labeled data (orange) . The dotted lines and colored areas show the mean and range the results obtained across 20 random trails.

If we set $x^* \in \mathbb{R}^n$ as the optimal cost with the i-th entry representing best expected cost starting from node i, we use the Bellman equation (see [40])

$$x_i^* = \min\left\{\mathbb{E}[C_{ij}^1 + x_j^*], \mathbb{E}[C_{ij}^2 + x_j^*]\right\} = \min\left\{\left\langle u_i^1, c_i^1 + x^*\right\rangle, \left\langle u_i^2, c_i^2 + x^*\right\rangle\right\}$$

and to formulate the stochastic shortest path as a deterministic optimization problem:

$$\min_x \sum_{i=1}^{d} \left| x_i - \min\left\{\left\langle u_i^1, x\right\rangle + v_i^1, \left\langle u_i^2, x\right\rangle + v_i^2\right\}\right| \tag{2.27}$$

where u_i^k is the i-th row of U^k and $v_i^k = \left\langle u_i^k, c_i^k\right\rangle$ with c_i^k the ith row of C^k for $k = 1, 2$. Problem (2.27) is nonsmooth and nonconvex; and using the method in the manuscript we write the approximate problem

$$\min_{x, w^1, w^2} h(w^1, w^2) + \frac{1}{2\nu}\left(\|A^1 x - w^1\|^2 + \|A^2 x - w^2\|^2\right) \tag{2.28}$$

where $A^k = U^k - I$, and $h(w^1, w^2) = \sum_{i=1}^d |\min\{w_i^1 + v_i^1, w_i^2 + v_i^2\}|$.

The optimal value of (2.28) is 0 because there is a solution to the Bellman equation. For the same reason, the solution of (2.27) and (2.28) coincide. The convergence results are shown in Figure 2.10, where we see a linear convergence rate in Figure 2.10. The obtained optimal policy is shown in Figure 2.11.

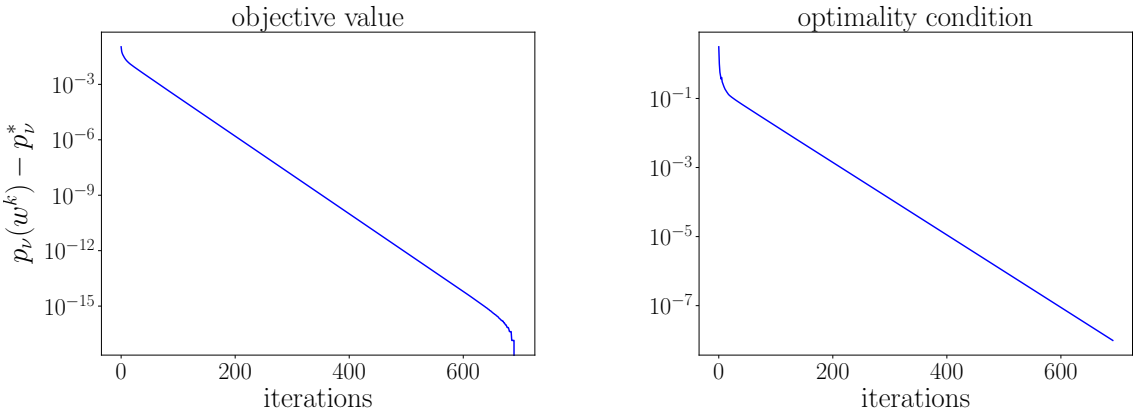


Figure 2.10: Convergence plot of stochastic shortest path experiment.

2.6.4 Convex and Nonconvex Clustering Problem

Clustering is a fundamental unsupervised learning technique. Basic approaches including k-means [183] and mixture models [110] are popular due to their simplicity and statistical interpretation. These approaches are built on essentially combinatorial subproblems (e.g. assigning members to clusters), making the approaches vulnerable to stalling at local minima. More recently, convex clustering formulations were proposed by [226] and [188].

The recent clustering formulations take the form

$$\min_X \ \frac{1}{2}\sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} \rho(x_i - x_j), \tag{2.29}$$

where $U = [u_1, \ldots, u_m]$ are the data points, $X = [x_1, \ldots, x_m]$ are the decision variables and ρ is the fusion regularizer. In the convex setting, ρ usually is chosen as the ℓ_2 norm, to encourage $x_i = x_j$; the number of different elements is controlled by the penalty λ. Problem (2.29) is then solved using splitting methods, including ADMM 4, or the alternating minimization algorithm (AMA) as proposed by [93]. The proposed RS approach is a natural competitor, especially given the results of Section 2.5.1.

Relaxing problem (2.29), we get the objective

$$\min_{x,w} \ \frac{1}{2}\sum_{i=1}^{m} \|x_i - u_i\|^2 + \lambda \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} \rho(w_{ij}) + \frac{1}{2\nu}\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} \|x_i - x_j - w_{ij}\|^2. \tag{2.30}$$

Algorithm (1) requires only a regularized least squares solve, and the proximal operator for ρ; it can be applied to both convex and nonconvex fusion penalties.

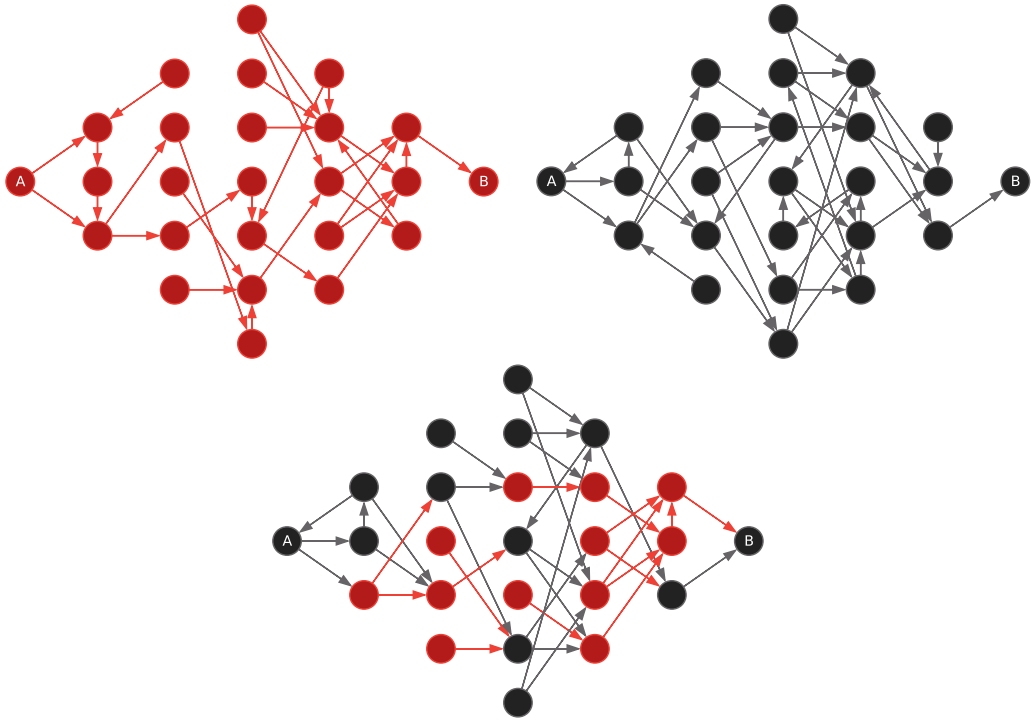


Figure 2.11: We want to move from node A to node B; and at each node we may switch between black and red graphs, shown in top left and top right panels, to minimize the expected cost. The optimal policy graph is shown in the bottom panel.

Comparison with ADMM. In this experiment, we generate a synthetic data set, with three clusters and 10 points per cluster. The hyper parameters are chosen as $\lambda = 0.5$ and $\nu = 1$. Results are shown in Figure 2.12, where we compare with ADMM and show the final adjacency matrix obtained from w_{ij}.

From the right plot of Figure 2.12, we can see that convex clustering via (2.29) and (2.30) cleanly identifies the clusters with these parameters. The left plot of Figure 2.12 shows identical performance between Algorithms 1 for (2.30) (blue) and ADMM for (2.29) (beige).

De-Biased Clustering. One issue with (2.30) is that $\rho = \|\cdot\|_2$ is very sensitive to λ, because of the bias introduced by points from different clusters. For this specific reason, we consider a nonconvex SCAD [142]-like regularizer,

$$\rho(d;\kappa) = \begin{cases} \|d\|, & \|d\| \leq \kappa \\ 0, & \|d\| > \kappa \end{cases}.$$

This regularizer allows us to use prior knowledge on the radius of each cluster, encoded by κ. This prior knowledge makes tuning λ easier, and also speeds up convergence of the clustering

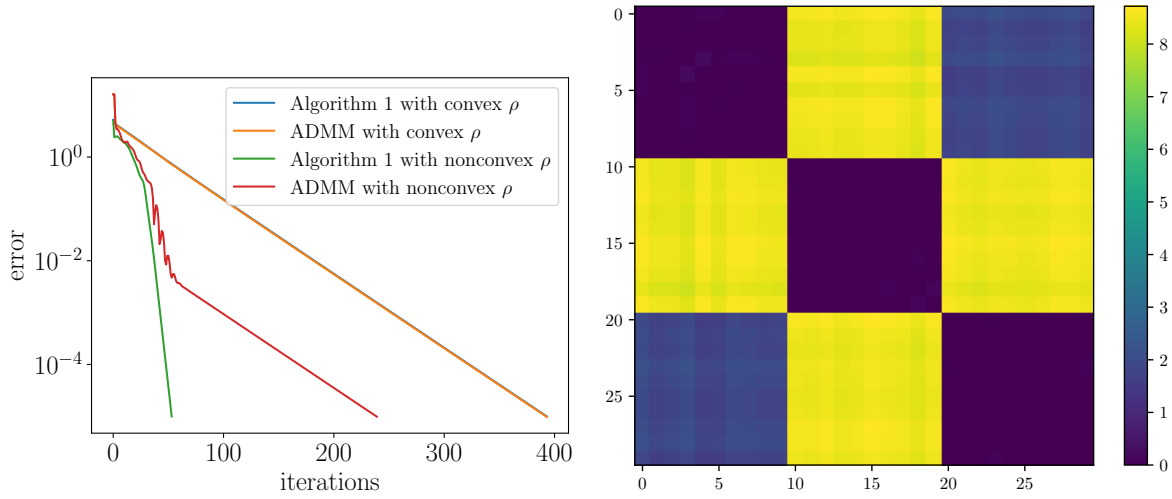


Figure 2.12: Clustering results. Left: convergence plots of Algorithm 1 with convex ρ (blue), ADMM with convex ρ (orange), Algorithm 1 with nonconvex ρ (green) and ADMM with nonconvex ρ (red). Right: adjacency matrix of the final results from Algorithm 1.

algoirthms. There is no convergence guarantee for ADMM when the SCAD penalty is used; however it still converges, even faster than for the convex case. Algorithm (1) is guaranteed to converge for (2.30), and has a significantly faster rate, see the left plot of Figure 2.12.

To test behavior with respect to the fusion penalty λ, we allow λ to vary in a grid from 0 to 1, and plot the path of the variables x_i. We also compare the convergence results for $\lambda = 0.5$ between convex and nonconvex ρ. These results are shown in Figure 2.13.

When we use clustering fusion penalties, all points affect one another; for larger values of the penalty λ, all points are rapidly assigned to a single cluster with center given by the center of mass of the point cloud. In contrast, using the nonconvex SCAD allows clusters that are far enough away to not affect each other, allowing desirable clustering behavior locally without the overall global effect.

2.7 Discussion

We have developed a new 'relax and split' approach for nonconvex-composite problems, and extended it to trimmed robust formulations. The approach applies to highly nonconvex models (those that are not even weakly convex), and can be easily applied to difficult structured nonsmooth nonconvex problems. The problem class is more general than those analyzed by recent sub-gradient based methods for nonsmooth nonconvex optimization.

We have also shown how the model and associated algorithms can be used for a variety of applications, including exact phase retrieval, semi-supervised classification, stochastic

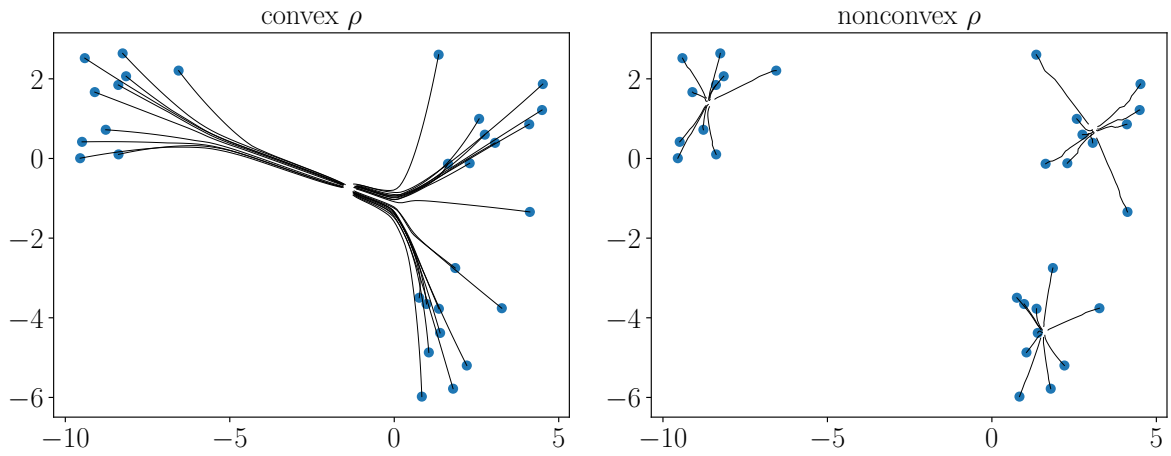


Figure 2.13: Comparison of the clustering paths for convex vs. nonconvex ρ across penalty parameters. Left: clustering path with convex $\rho = \|\cdot\|_2$. Right: clustering path of the variables using the nonconvex SCAD penalty ρ. Nonconvex fusion penalties give additional modeling flexibility and interpretable results.

shortest path problems, and new approaches to clustering. Every such application can be 'robustified' with the trimming extension, as we showed using the outlier-contaminated phase retrieval problem.

The paper opens several new avenues and raises important questions for future work, including a comprehensive analysis of inexact 'relax-and-split' approaches, extensions to compositions of nonconvex losses with nonlinear maps, and substantial detailed numerical work to evaluate the approach across a range of application domains.

2.8 Appendix

Theorem 1. Observe that,

$$\min_x g(x) + \frac{1}{2\nu}\|Ax - w\|^2 = \min_{x,y}\left\{g(x) + \frac{1}{2\nu}\|y - w\|^2 : y = Ax\right\}$$

Define $Ag(y) = \min_x\{g(x) : Ax = y\}$ which is the image of g under A. From [293] [Theorem 5.7] we know that Ag is a convex function. Moreover, since g is proper and bounded below, we know that Ag is also proper.

We cannot show Ag is closed unless we know more information about g and A [293] [Theorem 9.2]. Instead we show that for every w,

$$g_\nu(w) = \tilde{g}_\nu(w) := \min_x(\operatorname{cl} Ag)(y) + \frac{1}{2\nu}\|y - w\|^2,$$

where cl denotes the closure of the function.

Since $(\operatorname{cl} Ag)(y) \leq Ag(y)$ for all y, we know that,

$$g_\nu(w) \geq \tilde{g}_\nu(w).$$

Since $(\operatorname{cl} Ag)(y) + \frac{1}{2\nu}\|y-w\|^2$ is closed and strongly convex, we also know that there exist a unique minimizer,

$$y^* = \operatorname*{argmin}_y \ (\operatorname{cl} Ag)(y) + \frac{1}{2\nu}\|y-w\|^2.$$

From [293][Theorem 7.5], we know, for some $z \in \operatorname{ri} \operatorname{dom} Ag$,

$$\operatorname{cl} Ag(y^*) = \lim_{\lambda \uparrow 1} Ag(\lambda y^* + (1-\lambda)z).$$

Define the sequence $\{y_\lambda\}$, such that, $y_\lambda = \lambda y^* + (1-\lambda)z$. Since $y^* \in \operatorname{dom} \operatorname{cl} Ag = \operatorname{cl} \operatorname{dom} Ag$, using [293][Theorem 6.1] we know that for every $0 \leq \lambda < 1$, $y_\lambda \in \operatorname{ri} \operatorname{dom} Ag$. Therefore,

$$\tilde{g}_\nu(w) = Ag(y^*) + \frac{1}{2\nu}\|y^* - w\|^2 = \lim_{\lambda \uparrow 1} Ag(y_\lambda) + \frac{1}{2\nu}\|y_\lambda - w\|^2 \geq g_\nu(w),$$

so $g_\nu(w) = \tilde{g}_\nu(w)$. From [295] [Theorem 2.26] we know that $g_\nu(w)$ is a closed convex function, with a $\frac{1}{\nu}$-Lipschitz continuous gradient,

$$\nabla g_\nu(w) = \nabla \tilde{g}_\nu(w) = \frac{1}{\nu}(w - y^*).$$

Since $y^* \in \operatorname{dom} \operatorname{cl} Ag = \operatorname{cl} \operatorname{dom} Ag \subset \operatorname{Range}(A)$, we define $x^* = \{x : Ax = y^*\}$. Then we have the desired result,

$$\nabla g_\nu(w) = \frac{1}{\nu}(w - Ax), \quad \forall x \in x^*.$$

□

Theorem 2. Using the iteration of Algorithm 1, and introducing the sequence $\{x^k\}$, we have,

$$0 \in \frac{1}{\nu}A^\top(Ax^k - w^k) + \partial g(x^k), \quad 0 \in \frac{1}{\nu}(w^k - Ax^k) + \partial h(w^k).$$

From the definition of the objective, we have,

$$\begin{aligned}
p_\nu(w^k) =& h(w^k) + \frac{1}{2\nu}\|Ax^k - w^k\|^2 + g(x^k) \\
=& h(w^k) + \frac{1}{2\nu}\|Ax^{k-1} - w^k + A(x^k - x^{k-1})\|^2 + g(x^k) \\
=& h(w^k) + \frac{1}{2\nu}\|Ax^{k-1} - w^k\|^2 + \frac{1}{\nu}\left\langle Ax^{k-1} - w^k, A(x^k - x^{k-1})\right\rangle \\
&+ \frac{1}{2\nu}\|A(x^k - x^{k-1})\|^2 + g(x^k) \\
\leq& h(w^{k-1}) + \frac{1}{2\nu}\|Ax^{k-1} - w^{k-1}\|^2 + g(x^{k-1}) \\
&+ \frac{1}{\nu}\left\langle Ax^{k-1} - w^k, A(x^k - x^{k-1})\right\rangle + \frac{1}{2\nu}\|A(x^k - x^{k-1})\|^2 + g(x^k) - g(x^{k-1}).
\end{aligned}$$

Since g is convex,

$$g(x^k) - g(x^{k-1}) \leq \left\langle \partial g(x^k), x^k - x^{k-1} \right\rangle = \frac{1}{\nu} \left\langle w^k - Ax^k, A(x^k - x^{k-1}) \right\rangle$$

Therefore we have,

$$\begin{aligned} p_\nu(w^k) - p_\nu(w^{k-1}) \leq & \frac{1}{\nu} \left\langle Ax^{k-1} - w^k, A(x^k - x^{k-1}) \right\rangle + \frac{1}{2\nu} \|A(x^k - x^{k-1})\|^2 \\ & + \frac{1}{\nu} \left\langle w^k - Ax^k, A(x^k - x^{k-1}) \right\rangle \\ = & - \frac{1}{2\nu} \|A(x^{k-1} - x^k)\|^2 \end{aligned}$$

Summing up, we get

$$\frac{1}{k} \sum_{i=1}^{k} T_\nu(w^k) \leq \frac{1}{k} \sum_{i=1}^{k} \|\tfrac{1}{\nu} A(x^{i-1} - x^i)\|^2 \leq \frac{2}{\nu k} [p_\nu(w^0) - p_\nu^*],$$

as required. □

Lemma 1. *Define a sequence $d^k = \frac{1}{\nu}(w^k - w^{k+1})$ based on the iterates generated by Algorithm 1. If Assumption 3 holds, then p_ν has a minimizer w^*, and*

$$\left\langle w^k - w^*, d^k \right\rangle \geq \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \|\nu d^k\|^2 - \frac{1}{2\nu} \|\nu(I - P_A) d^k\|^2 + \frac{\alpha}{2} \|w^{k+1} - w^*\|^2.$$

Lemma 1.

$$\begin{aligned} p_\nu(w^{k+1}) &= \frac{1}{2\nu} \|(I - P_A) w^{k+1}\|^2 + h(w^{k+1}) \\ &= \frac{1}{2\nu} \|(I - P_A)(w^k - \nu d^k)\|^2 + h(w^{k+1}) \\ &= \frac{1}{2\nu} \|(I - P_A) w^k\|^2 - \frac{1}{\nu} \left\langle \nu d^k, (I - P_A) w^k \right\rangle + \frac{1}{2\nu} \|\nu(1 - P_A) d^k\|^2 + h(w^{k+1}) \end{aligned}$$

Decompose the first term above as follows:

$$\begin{aligned} \frac{1}{2\nu} \|(I - P_A) w^k\|^2 &= \frac{1}{2\nu} \|(I - P_A)(w^k - w^* + w^*)\|^2 \\ &= \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \left\langle w^*, (I - P_A)(w^k - w^*) \right\rangle + \frac{1}{2\nu} \|(I - P_A) w^*\|^2 \\ &= -\frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \left\langle w^k - w^*, (I - P_A) w^k \right\rangle + \frac{1}{2\nu} \|(I - P_A) w^*\|^2 \end{aligned}$$

Then we have,

$$\begin{aligned} p_\nu(w^{k+1}) = & - \frac{1}{2\nu} \|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu} \left\langle w^{k+1} - w^*, (I - P_A) w^k \right\rangle \\ & + \frac{1}{2\nu} \|(I - P_A) w^*\|^2 + \frac{1}{2\nu} \|\nu(1 - P_A) d^k\|^2 + h(w^{k+1}) \end{aligned}$$

Since h is convex and we know $d^k - \frac{1}{\nu}(w^k - P_A w^k) \in \partial h(w^{k+1})$ we have,

$$h(w^{k+1}) \leq h(w^*) + \frac{1}{\nu}\left\langle \nu d^k - (I - P_A)w^k, w^{k+1} - w^* \right\rangle - \frac{\alpha}{2}\|w^{k+1} - w^*\|^2$$

Combining these results, we get

$$\begin{aligned} p_\nu(w^{k+1}) \leq & -\frac{1}{2\nu}\|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu}\left\langle w^{k+1} - w^*, \nu d^k \right\rangle \\ & + \frac{1}{2\nu}\|(I - P_A)w^*\|^2 + \frac{1}{2\nu}\|\nu(1 - P_A)d^k\|^2 + h(w^*) - \frac{\alpha}{2}\|w^{k+1} - w^*\|^2 \\ 0 \leq p_\nu(w^{k+1}) - p_\nu(w^*) \leq & -\frac{1}{2\nu}\|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu}\left\langle w^k - w^*, \nu d^k \right\rangle \\ & - \frac{1}{\nu}\|\nu d^k\|^2 + \frac{1}{2\nu}\|\nu(1 - P_A)d^k\|^2 - \frac{\alpha}{2}\|w^{k+1} - w^*\|^2 \end{aligned}$$

which show the result:

$$\left\langle w^k - w^*, d^k \right\rangle \geq \frac{1}{2\nu}\|(I - P_A)(w^k - w^*)\|^2 + \frac{1}{\nu}\|\nu d^k\|^2 - \frac{1}{2\nu}\|\nu(1 - P_A)d^k\|^2 + \frac{\alpha}{2}\|w^{k+1} - w^*\|^2.$$

□

Theorem 5. Using the same $\{d^k\}$ as in Lemma 1,

$$\begin{aligned} \|w^{k+1} - w^*\|^2 &= \|w^k - \nu d^k - w^*\|^2 \\ \|w^{k+1} - w^*\|^2 &= \|w^k - w^*\|^2 - 2\left\langle w^k - w^*, \nu d^k \right\rangle + \|\nu d^k\|^2 \\ (1 + \alpha\nu)\|w^{k+1} - w^*\|^2 &\leq \|w^k - w^*\|^2 - \|(I - P_A)(w^k - w^*)\|^2 - \|\nu d^k\|^2 + \|\nu(I - P_A)d^k\|^2 \\ (1 + \alpha\nu)\|w^{k+1} - w^*\|^2 &\leq \|P_A(w^k - w^*)\|^2 - \|\nu P_A d^k\|^2 \\ \|w^{k+1} - w^*\|^2 &\leq \frac{1}{1 + \alpha\nu}\left(\|P_A(w^k - w^*)\|^2 - \|P_A(w^k - w^{k+1})\|^2\right) \\ \|w^{k+1} - w^*\|^2 &\leq \frac{1}{1 + \alpha\nu}\|w^k - w^*\|^2 \end{aligned}$$

□

Lemma 2. *If Assumption 4 holds, the iterates generated by the Algorithm 1 satisfy,*

$$\|P_A(w^k - w^{k+1})\| \leq \|P_A(w^k - w^*)\| \quad \forall k \in \mathbb{N}_+.$$

Lemma 2. Since $w^{k+1} = \operatorname{argmin}_w h(w) + \frac{1}{2\nu}\|w - P_A w^k\|^2$, we know,

$$h(w^{k+1}) + \frac{1}{2\nu}\|w^{k+1} - P_A w^k\|^2 \leq h(w^*) + \frac{1}{2\nu}\|w^* - P_A w^k\|^2.$$

By re-arranging terms, we get

$$\|w^{k+1} - P_A w^k\|^2 - \|(I - P_A)w^{k+1}\|^2 - (\|w^* - P_A w^k\|^2 - \|(I - P_A)w^*\|^2) \leq 2\nu(g_\nu(w^*) - g_\nu(w^{k+1})) \leq 0.$$

Since,

$$\|(I - P_A)w\|^2 + \|P_A(w - w^k)\|^2 = \|w - P_A w^k\|^2$$

we have,

$$\|w^{k+1} - P_A w^k\|^2 - \|(I - P_A)w^{k+1}\|^2 = \|P_A(w^k - w^{k+1})\|^2$$
$$\|w^* - P_A w^k\|^2 - \|(I - P_A)w^*\|^2 = \|P_A(w^k - w^*)\|^2$$

Therefore,

$$\|P_A(w^k - w^{k+1})\| \leq \|P_A(w^k - w^*)\| \quad \forall k \in \mathbb{N}_+.$$

□

Lemma 3. *Assume Assumption 4 holds, the iterates generated by the Algorithm 1 satisfy,*

$$\|w^{k+1} - w^*\|^2 \leq \|P_A(w^k - w^*)\|^2 - \|\nu P_A d^k\|^2.$$

Moreover,

$$\|w^{k+1} - w^*\| \leq \|P_A(w^k - w^*)\|.$$

Lemma 3. The proof uses the same technique as the proof of Theorem 5. □

Theorem 6. Since $w^{k+1} = \operatorname{argmin}_w h(w) + \frac{1}{2\nu}\|w - P_A w^k\|^2$, we know,

$$\begin{aligned} 0 &\in \partial h(w^{k+1}) + \frac{1}{\nu}(w^{k+1} - P_A w^k) \\ \frac{1}{\nu}P_A(w^k - w^{k+1}) &\in \partial h(w^{k+1}) + \frac{1}{\nu}(w^{k+1} - P_A w^{k+1}) \\ \frac{1}{\nu}P_A(w^k - w^{k+1}) &\in \partial p_\nu(w^{k+1}) \end{aligned}$$

Because p_ν is convex and w^* is a sharp minima,

$$\alpha\|w^{k+1} - w^*\| \leq p_\nu(w^{k+1}) - p_\nu(w^*) \leq \frac{1}{\nu}\left\langle P_A(w^k - w^{k+1}), w^{k+1} - w^*\right\rangle$$

Expanding the right inequality we obtain

$$\begin{aligned} p_\nu(w^{k+1}) - p_\nu(w^*) &\leq \frac{1}{\nu}\left\langle P_A(w^k - w^{k+1}), w^{k+1} - w^k + w^k - w^*\right\rangle \\ &\leq -\frac{1}{\nu}\|P_A(w^k - w^{k+1})\|^2 + \frac{1}{\nu}\left\langle P_A(w^k - w^{k+1}), w^k - w^*\right\rangle \\ &\leq \frac{1}{\nu}\|P_A(w^k - w^{k+1})\|\|w^k - w^*\| \\ &\leq \frac{1}{\nu}\|P_A(w^k - w^*)\|\|w^k - w^*\| \\ &\leq \frac{1}{\nu}\|w^k - w^*\|^2 \end{aligned}$$

Therefore,

$$\|w^{k+1} - w^*\| \leq \frac{1}{\alpha\nu}\|w^k - w^*\|^2.$$

Combined with Lemma 3 we have, for all $k \geq K$,

$$\|w^{k+1} - w^*\| \leq \min\left\{\|w^k - w^*\|, \frac{1}{\alpha\nu}\|w^k - w^*\|^2\right\}$$

which gives the locally quadratic convergence rate. □

Theorem 7. We introduce a sequence $\{x^k\}$ that statisfies,

$$x^k = \operatorname*{argmin}_x \frac{1}{2\nu}\|Ax - w^k\| + g(x), \quad A^\top(w^k - Ax^k) \in \nu\partial g(x^k), \quad \nu\nabla g_\nu(w^k) = w^k - Ax^k.$$

Then we know the iterates of Algorithm 3 satisfy,

$$\begin{aligned}\frac{1}{\nu}A(x^k - x^{k+1}) &\in \nabla g_\nu(w^{k+1}) + \sum_{i=1}^m v_i^k \partial h_i(w^{k+1}),\\ \frac{1}{\alpha}(v^k - v^{k+1}) &\in H(w^{k+1}) + \partial\delta(v^{k+1}|\triangle_\tau).\end{aligned}$$

By definition we know,

$$\begin{aligned}p_\nu^t(w^{k+1}, v^k) =& \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + g_\nu(w^{k+1})\\ =& \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + \frac{1}{2\nu}\|Ax^{k+1} - w^{k+1}\|^2 + g(x^{k+1})\\ =& \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + \frac{1}{2\nu}\|Ax^k - w^{k+1} + A(x^{k+1} - x^k)\|^2 + g(x^{k+1})\\ =& \sum_{i=1}^m v_i^k h_i(w_i^{k+1}) + \frac{1}{2\nu}\|Ax^k - w^{k+1}\|^2\\ &+ \frac{1}{\nu}\left\langle Ax^k - w^{k+1}, A(x^{k+1} - x^k)\right\rangle + \frac{1}{2\nu}\|A(x^{k+1} - x^k)\|^2 + g(x^{k+1})\\ \leq& \sum_{i=1}^m v_i^k h_i(w_i^k) + \frac{1}{2\nu}\|Ax^k - w^k\|^2\\ &+ \frac{1}{\nu}\left\langle Ax^k - w^{k+1}, A(x^{k+1} - x^k)\right\rangle + \frac{1}{2\nu}\|A(x^{k+1} - x^k)\|^2 + g(x^{k+1})\end{aligned}$$

Since g is convex, we have,

$$\begin{aligned} g(x^k) &\geq g(x^{k+1}) + \frac{1}{\nu}\left\langle A^\top(w^k - Ax^k), x^k - x^{k+1}\right\rangle \\ &= g(x^{k+1}) + \frac{1}{\nu}\left\langle w^k - Ax^k, A(x^k - x^{k+1})\right\rangle. \end{aligned}$$

Plug this inequality into the result above, we get

$$p_\nu^t(w^{k+1}, v^k) \leq \sum_{i=1}^m v_i^k h_i(w_i^k) + \frac{1}{2\nu}\|Ax^k - w^k\|^2 + g(x^k) - \frac{1}{2\nu}\|A(x^k - x^{k+1})\|^2,$$

$$p_\nu^t(w^{k+1}, v^k) - p_\nu^t(w^k, v^k) \leq -\frac{1}{2\nu}\|A(x^k - x^{k+1})\|^2.$$

An analogous calculation for v gives us

$$\begin{aligned} &p_\nu^t(w^{k+1}, v^{k+1}) - p_\nu^t(w^{k+1}, v^k) \\ =&\left\langle H(w^{k+1}), v^{k+1} - v^k\right\rangle + \delta(v^{k+1}|\triangle_\tau) - \delta(v^k|\triangle_\tau) \\ =&-\frac{1}{\alpha}\|v^{k+1} - v^k\|^2 - \left[\delta(v^k|\triangle_\tau) - \left(\delta(v^{k+1}|\triangle_\tau) + \left\langle \partial\delta(v^{k+1}|\triangle_\tau), v^k - v^{k+1}\right\rangle\right)\right] \\ \leq&-\frac{1}{\alpha}\|v^{k+1} - v^k\|^2 \end{aligned}$$

Therefore we can conclude that,

$$\begin{aligned} T_\nu^t(w^{k+1}, v^{k+1}) &\leq \frac{1}{2\nu}\|A(x^k - x^{k+1})\|^2 + \frac{1}{\alpha}\|v^{k+1} - v^k\|^2 \\ &\leq p_\nu^t(w^k, v^k) - p_\nu^t(w^{k+1}, v^k) + p_\nu^t(w^{k+1}, v^k) - p_\nu^t(w^{k+1}, v^{k+1}) \\ &= p_\nu^t(w^k, v^k) - p_\nu^t(w^{k+1}, v^{k+1}) \end{aligned}$$

Adding up the telescoping series, we get the final result:

$$\frac{1}{k}\sum_{i=1}^k T_\nu^t(w^i, v^i) \leq \frac{1}{k}[p_\nu^t(w^0, v^0) - p_\nu^t(w^k, v^k)].$$

□

Chapter 3

A UNIFIED FRAMEWORK FOR SPARSE RELAXED REGULARIZED REGRESSION

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 such as total variation (TV) and its nonconvex variants. 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. To promote reproducible research, we also provide a companion MATLAB package that implements these examples.

3.1 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, outliers, etc). 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 on 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 [220] and Gauss [157, 158], 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, resulting in 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. Andrey Tikhonov [331] 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

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda R(\mathbf{C}\boldsymbol{x})\,, \tag{3.1}$$

where $\boldsymbol{x}\in\mathbb{R}^d$ is the unknown signal, $\mathbf{A}\in\mathbb{R}^{m\times d}$ is the linear data-generating mechanism 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 30 years later [189]. 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 Chen and Donoho in 1994 [310] as *basis pursuit*, and by Tibshirani in 1996 [329] as the *least absolute shrinkage and selection operator* (LASSO). More generally, the ℓ_1 norm was introduced much earlier: as a penalty in 1969 [279], with specialized algorithms in 1973 [98], and as a robust loss in geophysics in 1973 [94]. In modern optimization, nonsmooth regularizers are widely used across a diverse set of applications, including in the training of neural network architectures [168]. Figure 3.1(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 $\mathbb{B}_1\leq\tau$. 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 3.2. We demonstrate the advantages of SR3 (computational efficiency, higher accuracy, faster convergence rates, greater flexibility) across a range of regularized regression problems with synthetic

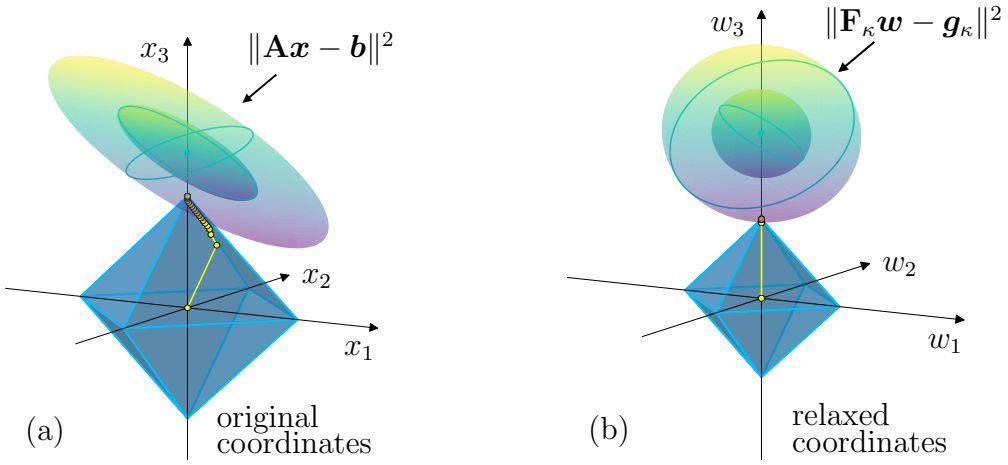


Figure 3.1: (a) Level sets (green ellipses) of the quadratic part of LASSO (3.1) and corresponding path of prox-gradient to the solution (40 iterations) in $\boldsymbol{x}$-coordinates. (b) Level sets (green spheres) of the quadratic part of the SR3 value function (3.3) and corresponding SR3 solution path (2 iterations) in relaxed coordinates $\boldsymbol{w}$. 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 $\mathbf{A}$ with a weaker effect on the small ones, 'squashing' the level sets into approximate spheres, accelerating convergence, and improving performance.

and real data, including applications in compressed sensing, LASSO, matrix completion, TV regularization, and group sparsity using a range of test problems in Section 3.3.

3.2 SR3 Method

Our goal is to improve the robustness, computational efficiency, and accuracy of sparse and nonsmooth formulations. We *relax* (3.1) 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 [383]. The general SR3 formulation modifies (3.1) to the following

$$\min_{\boldsymbol{x},\boldsymbol{w}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \lambda R(\boldsymbol{w}) + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x} - \boldsymbol{w}\|^2, \tag{3.2}$$

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) [142], and easily handles linear composite regularizers. Two widely used examples that rely on compositions are compressed sensing formulations that

use tight frames [117], and total variation (TV) regularization in image denoising [303].

In the convex setting, the formulation (3.2) fits into a class of problems studied by Bauschke, Combettes, and Noll [33], who credit the natural alternating minimization algorithm to Acker and Prestel in 1980 [2], and the original alternating projections method to Cheney and Goldstein in 1959 [92] and Von Neumann in 1950 [347, Theorem 13.7]. The main novelty of the SR3 approach is in using (3.2) to extract information from the $\boldsymbol{w}$ variable. We also allow nonconvex regularizers $R(\cdot)$, using the structure of (3.2) 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 (3.2), relieving the pressure these competing goals put on $\boldsymbol{x}$ in (3.1). Second, we can partially minimize (3.2) 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 3.1(b) illustrates this advantage of SR3 on the LASSO problem.

3.2.1 SR3 and Value Function Optimization

Associated with (3.2) is a *value function* formulation that allows us to precisely characterize the relaxed framework. Minimizing (3.2) in $\boldsymbol{x}$, we obtain the value function

$$v(\boldsymbol{w}) := \min_{\boldsymbol{x}} \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2 + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x}-\boldsymbol{w}\|^2. \tag{3.3}$$

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

$$\begin{aligned} \mathbf{F}_\kappa &= \begin{bmatrix} \kappa\mathbf{A}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \\ \sqrt{\kappa}(\mathbf{I}-\kappa\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top) \end{bmatrix}, & \mathbf{F}_\kappa &\in \mathbb{R}^{(m+n)\times n} \\ \mathbf{G}_\kappa &= \begin{bmatrix} \mathbf{I}-\mathbf{A}\mathbf{H}_\kappa^{-1}\mathbf{A}^\top \\ \sqrt{\kappa}\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{A}^\top \end{bmatrix}, & \mathbf{G}_\kappa &\in \mathbb{R}^{(m+n)\times m} \\ \boldsymbol{g}_\kappa &= \mathbf{G}_\kappa\boldsymbol{b}, & \boldsymbol{g}_\kappa &\in \mathbb{R}^{m+n} \end{aligned} \tag{3.4}$$

which gives a closed form for (3.3):

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

Problem (3.2) then reduces to

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\mathbf{F}_\kappa\boldsymbol{w}-\boldsymbol{g}_\kappa\|^2 + \lambda R(\boldsymbol{w})\ . \tag{3.5}$$

The ellipsoid in Fig. 3.1(a) shows the level sets of $\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2$, while the spheroid in Fig. 3.1(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 3.1, 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 $\mathrm{cond}(\mathbf{A}) := \sigma_{\max}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$ 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 $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$) [91, 117, 132].

Theorem 9. *When $\lambda = 0$, (3.5) and (3.1) share the same solution set. We also have the following relations:*

$$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa = \kappa\mathbf{I} - \kappa^2\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \tag{3.6}$$

$$\sigma_i(\mathbf{F}_\kappa^\top\mathbf{F}_\kappa) = \kappa - \kappa^2\sigma_{n-i+1}(\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top). \tag{3.7}$$

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

$$\mathbf{F}_\kappa^\top\mathbf{F}_\kappa = \mathbf{A}^\top(\mathbf{I} + \mathbf{A}\mathbf{A}^\top/\kappa)^{-1}\mathbf{A} \tag{3.8}$$

$$\sigma_i(\mathbf{F}_\kappa^\top\mathbf{F}_\kappa) = \frac{\sigma_i(\mathbf{A}^\top\mathbf{A})}{1 + \sigma_i(\mathbf{A}^\top\mathbf{A})/\kappa}\,, \tag{3.9}$$

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

$$\mathrm{cond}(\mathbf{F}_\kappa) = \mathrm{cond}(\mathbf{A})\sqrt{\frac{\kappa + \sigma_{\min}(\mathbf{A})^2}{\kappa + \sigma_{\max}(\mathbf{A})^2}}\,. \tag{3.10}$$

Theorem 9 lets us interpret (3.5) as a re-weighted version of the original problem (3.1). 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 9 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,*

$$\sigma_i(\mathbf{F}_\kappa) = \begin{cases} \sqrt{\kappa}\,\frac{\sigma_{i-(n-d)}(\mathbf{A})}{\sqrt{\kappa+\sigma_{i-(n-d)}(\mathbf{A})^2}} & i > n-d \\ \sqrt{\kappa} & i \leq n-d \end{cases}. \tag{3.11}$$

For $n > d$, this implies

$$\mathrm{cond}(\mathbf{F}_\kappa) = \mathrm{cond}(\mathbf{A})\sqrt{\frac{\kappa + \sigma_{\min}(\mathbf{A})^2}{\sigma_{\max}(\mathbf{A})^2}}\,. \tag{3.12}$$

When $n = d$, this implies

$$\operatorname{cond}(\mathbf{F}_\kappa) = \operatorname{cond}(\mathbf{A})\sqrt{\frac{\kappa + \sigma_{\min}(\mathbf{A})^2}{\kappa + \sigma_{\max}(\mathbf{A})^2}}\,. \tag{3.13}$$

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 9,

$$\bar{\mathbf{C}}^\top \mathbf{F}_\kappa^\top \mathbf{F}_\kappa \bar{\mathbf{C}} = \begin{bmatrix}\mathbf{A}^\top(\mathbf{I} + \mathbf{A}\mathbf{A}^\top/\kappa)^{-1}\mathbf{A} & \\ & \kappa\mathbf{I}_{n-d}\end{bmatrix}\,. \tag{3.14}$$

The result follows from the second part of Theorem 9 . □

When $\mathbf{C}$ is a square orthogonal matrix, partial minimization of (3.3) 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 3.1 for $\mathbf{C} = \mathbf{I}$. As a result, iterative methods for (3.5) converge much faster than the same methods applied to (3.1), especially for ill-conditioned $\mathbf{A}$. The geometry of the level sets of (3.5) also encourages the discovery of sparse solutions; see the path-to-solution for each formulation in Figure 3.1. 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, $\operatorname{cond}(\mathbf{F}_\kappa) \leq 1 + \operatorname{cond}(\mathbf{A})/\mu$.

3.2.2 Algorithms for the SR3 Problem

Problem (3.5) can be solved using a variety of algorithms, including the prox-gradient method detailed in Algorithm 5. In the convex case, Algorithm 5 is equivalent to the alternating method of [33]. The $\boldsymbol{w}$ update is given by

$$\hat{\boldsymbol{w}}^{k+1} = \operatorname{prox}_{\frac{\lambda}{\kappa}R}\left(\boldsymbol{w}^k - \frac{1}{\kappa}\mathbf{F}_\kappa^\top(\mathbf{F}_\kappa \boldsymbol{w}^k - \boldsymbol{g}_\kappa)\right)\,, \tag{3.15}$$

where $\operatorname{prox}_{\frac{\lambda}{\kappa}R}$ is the *proximity operator* (prox) for R (see e.g. [96]) evaluated at $\mathbf{C}\boldsymbol{x}$. The prox in Algorithm 5 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, $\operatorname{prox}_{\lambda\|\cdot\|_1}$ is the *soft-thresholding* (ST) operator:

$$\operatorname{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{x})_i = \operatorname{sign}(x_i)\max(|x_i| - \lambda, 0). \tag{3.16}$$

Algorithm 5 is the proximal gradient algorithm applied to (3.5). It is useful to contrast it with the proximal gradient algorithm for the original problem (3.1), detailed in Algorithm 6. First, Algorithm 6 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

$$\operatorname{prox}_{\lambda\|\mathbf{C}\cdot\|_1}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}} \frac{1}{2\lambda}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \|\mathbf{C}\boldsymbol{y}\|_1. \tag{3.17}$$

Algorithm 5 SR3 for (3.2)

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

Algorithm 6 Prox-gradient for (3.1)

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

In contrast, Algorithm 1 always solves (3.5), which is regularized by $R(\boldsymbol{w})$ rather than a composition, with $\mathbf{C}$ affecting $\mathbf{F}_\kappa$ and $\boldsymbol{g}_\kappa$, see (3.4). This simple observation has important consequences, since the prox-gradient method converges for a wide class of problems, including non-convex regularizers [23]. For regularized least squares problems specifically, we derive a self-contained convergence theorem with a sublinear convergence rate.

Theorem 10 (Proximal Gradient Descent for Regularized Least Squares)**.** *Consider the linear regression objective,*

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

where p is bounded below, so that

$$-\infty < p^* = \inf_{\boldsymbol{x}} p(\boldsymbol{x}),$$

and R may be nonsmooth and nonconvex. With step $t = 1/\sigma_{\max}(\mathbf{A})^2$, the iterates generated by Algorithm 6 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*

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

Therefore Algorithm 6 converges at a sublinear rate to a stationary point of p*.*

Theorem 10 always applies to the SR3 approach, which uses value function (3.5). When $\mathbf{C} = \mathbf{I}$, we can also compare the convergence rate of Algorithm 5 for (3.5) to the rate for Algorithm 6 for (6). In particular, the rates of Algorithm 5 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 11.

Theorem 11. *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 6 applied to* p_x*, and* $\boldsymbol{w}_k$ *denote the iterates of Algorithm 5 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:*

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

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

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

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

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

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

When $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, algorithm 6 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 $n > d$.

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 5 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:

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

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

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

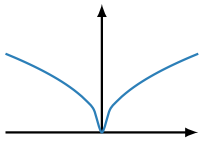
For convex R and $\mathbf{A}$ with full rank, we also have

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

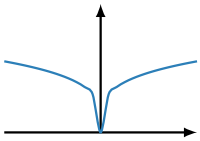
Algorithm 5 can be used with both convex and nonconvex regularizers, as long as the prox operator of the regularizer is available. A growing list of proximal operators is reviewed by [96]. Notable nonconvex prox operators in the literature include (1) indicator of set of rank r matrices, (2) spectral functions (with proximable outer functions) [124, 222], (3) indicators of unions of convex sets (project onto each and then choose the closest point), (4) MCP penalty [376], (5) firm-thresholding penalty [156], 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.

(a) ℓ_0 norm.

(b) Clipped absolute deviation.



(c) ℓ_p norm ($p = \frac{1}{2}$).



(d) ℓ_p norm ($p = \frac{1}{4}$).

Figure 3.2: Nonconvex sparsity promoting regularizers.

3.2.3 Nonconvex Regularizers and Constraints

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 5.1. The nonconvex ℓ_0 has a simple prox — hard thresholding (HT) [50], see Table 3.1. The SR3 formulation with the ℓ_0 regularizer uses HT instead of the ST operator (3.16) in line 5 of Algorithm 5.

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. [212] and the references within. Two members of this family are shown in panels (c) and (d) of Figure 5.1. The ℓ_p^p prox subproblem is given by

$$\min_x \; f_{\alpha,p}(x; z) := \frac{1}{2\alpha}(x - z)^2 + |x|^p \tag{3.18}$$

This problem is studied in detail by [89]. 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 (3.18) [89, Proposition 8]. Our implementation is summarized in the Appendix. The $\ell_{1/2}$ regularizer is particularly important for CS, and is known to do better than either ℓ_0 or ℓ_1.

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

Table 3.1: Proximal operators of sparsity-promoting regularizers.

Nonconvex Regularizers: (S)CAD

The (Smoothly) Clipped Absolute Deviation (SCAD) [142] 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 5.1, and the analytic prox is given in Table 3.1. This regularizer, when combined with SR3, obtains the best results in the CS experiments in Section 3.3.

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 (3.23)). The SR3 formulation is solved by Algorithm 5, a prox-gradient (primal) method. In contrast, most TV algorithms use primal-dual methods because of the composition $\|\mathbf{C}\boldsymbol{x}\|_1$ [84].

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 ($\boldsymbol{x}\ge 0$) and nonconvex spherical constraints ($\|\boldsymbol{x}\|_2 = r$).

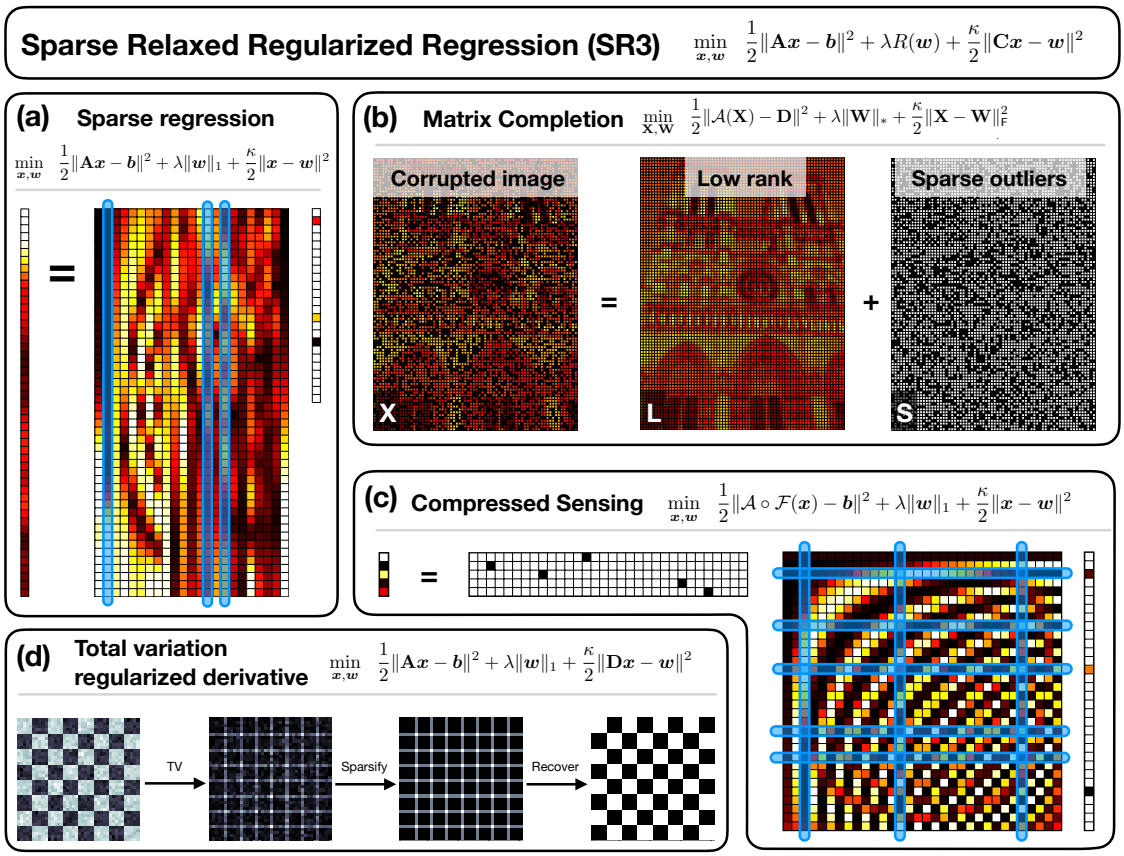


Figure 3.3: Common optimization applications where the SR3 method improves performance. For each method, the specific implementation of our general architecture (3.2) is given.

3.2.4 Optimality of SR3 Solutions

We now consider the relationship between the optimal solution $\hat{\boldsymbol{w}}$ to problem (3.5), and the original problem (3.1).

Theorem 12 (Optimal Ratio). *Assume* $\mathbf{C} = \mathbf{I}$, *and let* λ_1 *for* (3.1) *and* λ_2 *for* (3.5) *be related by the ratio* $\tau = \lambda_2/\lambda_1$, *and let* $\hat{\boldsymbol{w}}^k$ *be the optimal solution for* (3.5) *with parameter* λ_2. *If* λ_2 *is set to be* $\tau\lambda_1$ *where*

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

then have that the distance to optimality of $\hat{\boldsymbol{w}}^1$ *for* (3.1) *is bounded above by*

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

Theorem 12 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 (3.1), and characterizes the distance of $\hat{\boldsymbol{w}}$ to optimality of the original problem. The proof is given in the Appendix.

Theorem 12 shows that as κ increases, the solution $\hat{\boldsymbol{w}}$ moves closer to being optimal for the original problem (3.1). On the other hand, Theorem 11 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 3.3.

3.3 Results

The formulation (3.1) covers many standard problems, including variable selection (LASSO), compressed sensing, TV-based image de-noising, and matrix completion, shown in Fig. 3.3. 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.

3.3.1 SR3 vs. LASSO and Compressed Sensing

Using Eqs. (3.1) and (3.2), the LASSO and associated SR3 problems are

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{x}\|_1 \tag{3.19}$$

$$\min_{\boldsymbol{x},\boldsymbol{w}} \quad \frac{1}{2}\|\mathbf{A}\boldsymbol{x}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{w}\|_1+\frac{\kappa}{2}\|\boldsymbol{x}-\boldsymbol{w}\|^2 \tag{3.20}$$

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. (3.19) and (3.20). The formulation in (3.20) is related to an earlier sequentially thresholded least square algorithm that was used for variable selection to identify nonlinear dynamical systems from data [61].

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.

LASSO Path

The LASSO path refers to the set of solutions obtained by sweeping over λ in (3.1) from a maximum λ, which gives $\boldsymbol{x}=\mathbf{0}$, down to $\lambda=0$, which gives the least squares solution. In [320], it was shown that (3.19) makes mistakes early along this path.

Problem setup. As in [320], 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

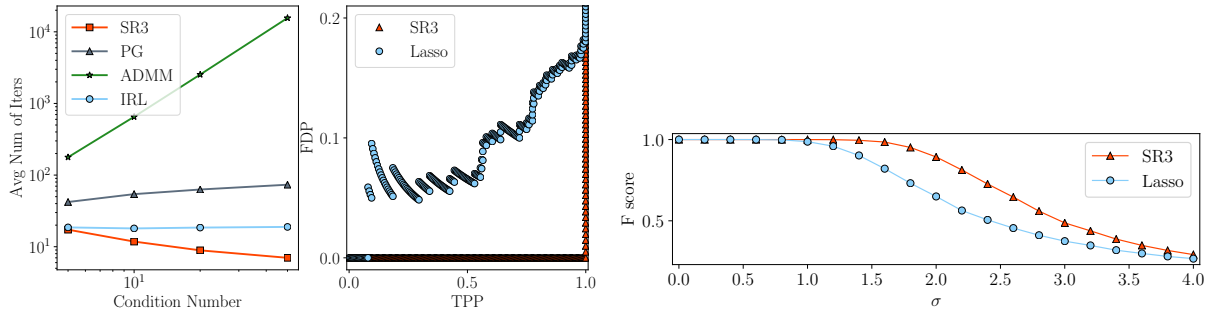


Figure 3.4: **Left:** SR3 approach (red) is orders of magnitude faster than ADMM (green) or other first-order methods such as prox-gradient (gray). While IRL (blue) requires a comparable number of iterations, its cost per iteration is more expensive than SR3. **Middle:** True Positives vs. False Positives along the LASSO path (blue) and along the SR3 path (red). **Right:** F_1 score of SR3 (red) and LASSO formulation (blue) with respect to different noise levels.

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. 3.4. LASSO makes mistakes early along the path [320]. In contrast, SR3 recovers the support without introducing any false positives along the entire path until overfitting sets in with the 201^{st} nonzero entry.

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.

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 (3.19) and (3.20) 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 (3.19) and (3.20) 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. 3.4. SR3 has a uniformly higher F_1-score across all noise levels.

Computational Efficiency.

We compare the computational efficiency of the Alternating Directions Method of Multipliers (ADMM) (see e.g. [55, 162]), proximal gradient algorithms (see e.g. [96]) on (3.19) with Algorithm 5, and a state-of-the-art Iteratively Reweighted Least-Squares (IRL) method, specifically IRucLq-v as in [212].

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.

Parameter selection. We choose $\kappa = 1$, λ in (3.19) to be $\|\mathbf{A}^\top \boldsymbol{b}\|_\infty/5$, and λ in (3.20) to be $\|\mathbf{F}_\kappa^\top \boldsymbol{g}_\kappa\|_\infty/5$.

Table 3.2: 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	$O(mn^2 + n^3)$	$O(n^2)$

Results. The results (by number of iterations) are shown in the top left panel of Fig. 3.4. The complexity of each iteration is given in Table 3.2. 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. The IRucLq-v method requires the formation and inversion of such a matrix at each iteration. From Fig. 3.4, SR3 requires far fewer iterations than ADMM and the proximal gradient method, especially as $\text{cond}(\mathbf{A})$ increases. SR3And the IRucLq-v method require a comparable number of iterations. A key difference is that ADMM requires dual variables, while SR3 is fundamentally a primal-only method. When $\text{cond}(\mathbf{A}) = 50$, ADMM needs almost 10^4 iterations to solve (3.19); proximal gradient descent requires 10^2 iterations; and SR3 requires 10 to solve (3.20). Overall, the SR3 method takes by far the least total compute time as the condition number increases. More detailed experiments, including for larger systems where iterative methods are needed, are left to future work.

SR3 for Compressed Sensing.

When $m \ll n$, the variable selection problem targeted by (3.19) is often called *compressed sensing* (CS). Sparsity is required to make the problem well-posed, as (3.19) 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 [80]. 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 [82, 80, 81, 117, 116] to analyze and understand the behavior of (3.1) 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) [82] — 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, 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 [82] 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 (3.1) and (3.2) 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 5, 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 3.5, for relatively incoherent random Gaussian measurements, both the standard formulation (3.1) 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 (3.1) degrade drastically. In this case, SR3 is the most effective approach for each regularizer and provides

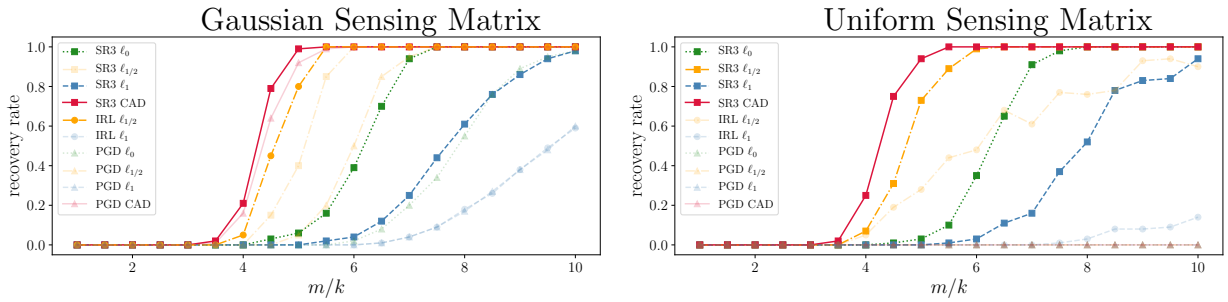


Figure 3.5: Compressed sensing results: recovering a 20-sparse signal in $\mathbb{R}^{500}$ from a small number of measurements. We plot the recovery rate as the number of measurements increases. Line color and style are determined by the regularizer while marker shapes are determined by the algorithm/formulation used. For readability, only the best performing algorithm for each regularizer is plotted in bold, with the rest opaque. **Left panel:** the sensing matrix $\mathbf{A}$ 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 $\mathbf{A}$ 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 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.

Remark: Many algorithms focus on the noiseless setting in compressive sensing, where the emphasis shifts to recovering signals that may have very small amplitudes [212]. 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.

Analysis vs. Synthesis

Compressive sensing formulations fall into two broad categories, analysis (3.21) and synthesis (3.22) (see [91, 132]):

$$\min_{\boldsymbol{x}} \ \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + R(\mathbf{C}\boldsymbol{x}), \tag{3.21}$$

$$\min_{\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{A}\mathbf{C}^\top\boldsymbol{\xi} - \boldsymbol{b}\|^2 + R(\boldsymbol{\xi}), \tag{3.22}$$

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 $n \leq d$, and very different when $n > d$ [91]. Both forms appear in a variety of inverse problems including denoising, interpolation and super-resolution. The work of [132] presents a thorough comparison of (3.21) and (3.22) 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 (3.21), 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 [194] for solving (3.21).

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. In this experiment, 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 3.6. 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 (3.21) identifies the

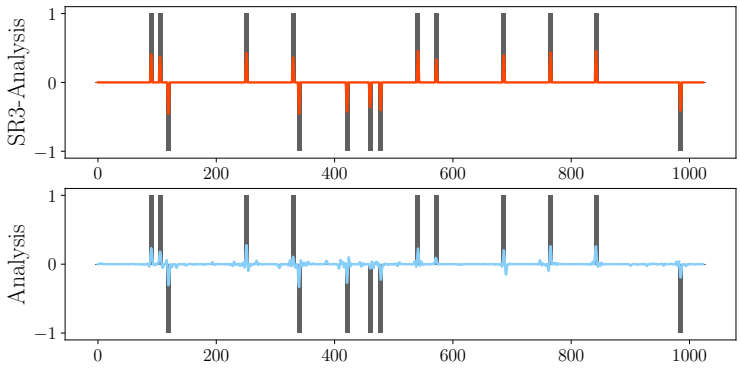


Figure 3.6: Comparison of standard analysis with SR3-analysis. **Top panel:** result using SR3-analysis, plotting the final $\boldsymbol{w}$ (red) against the true signal (dark grey). **Bottom panel:** result using standard analysis and the IRL-D algorithm, plotting final $\mathbf{C}\boldsymbol{x}$ (blue) against the true signal (dark grey).

support but is not completely sparse, requiring post-processing steps such as thresholding to get a support estimate.

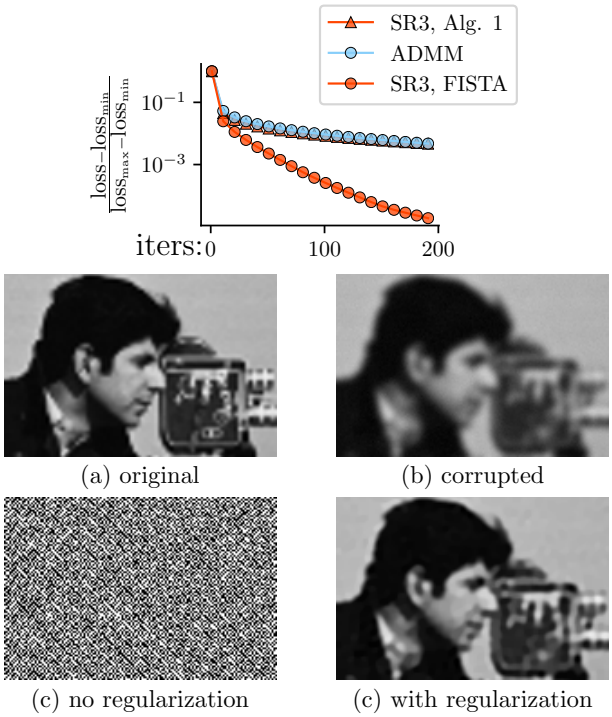
3.3.2 SR3 for Total Variation Regularization

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 [303, 85, 319, 268, 351, 36, 84]. 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.

$$R_{\text{TV}}\begin{pmatrix}\mathbf{D}_x\mathbf{X}\\ \mathbf{D}_y\mathbf{X}\end{pmatrix} := \sum_{i=1}^{m}\sum_{j=1}^{n}\sqrt{[\mathbf{D}_x\mathbf{X}]_{ij}^2 + [\mathbf{D}_y\mathbf{X}]_{ij}^2}\,. \tag{3.23}$$

Adding the TV regularizer (3.23) to a regression problem corresponds to imposing a sparsity prior on the discrete gradient.

Consider image deblurring (Fig. 3.7). 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



(a) original (b) corrupted

(c) no regularization (c) with regularization

Figure 3.7: The top plot compares the progress of the SR3 and ADMM-type algorithms in reducing their losses, showing similar rates of convergence. 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 [84] is visually similar, with a similar SNR)

is the noise level. To regularize the problem of recovering $\mathbf{X}$ from the corrupted signal $\mathbf{B}$, we add the TV regularization:

$$\hat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A} * \mathbf{X} - \mathbf{B}\|_F^2 + \lambda R_{\text{TV}} \begin{pmatrix} \mathbf{D}_x\mathbf{X} \\ \mathbf{D}_y\mathbf{X} \end{pmatrix} . \tag{3.24}$$

The natural SR3 reformulation is given by

$$\begin{aligned} \min_{\mathbf{X},\boldsymbol{w}_x,\boldsymbol{w}_y} & \frac{1}{2}\|\mathbf{A}*\mathbf{X}-\mathbf{B}\|_F^2 \\ & + \lambda R_{\text{TV}} \begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} + \frac{\kappa}{2} \left\| \begin{matrix} \boldsymbol{w}_x - \mathbf{D}_x\mathbf{X} \\ \boldsymbol{w}_y - \mathbf{D}_y\mathbf{X} \end{matrix} \right\|_F^2 . \end{aligned} \tag{3.25}$$

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 (3.24) or (3.25)).
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 3.7 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 7 FISTA for SR3 TV

Input: $\boldsymbol{w}^0$
Initialize: $k = 0$, $a_0 = 1$, $\boldsymbol{v}_0 = \boldsymbol{w}^0$, $\eta \leq \frac{1}{\kappa}$
while not converged **do**
  $k \leftarrow k + 1$
  $\boldsymbol{v}_k \leftarrow \text{prox}_{\eta R}(\boldsymbol{w}^{k-1} - \eta(\mathbf{F}_\kappa^\top(\mathbf{F}_\kappa \boldsymbol{w}^{k-1} - \boldsymbol{g}_\kappa)))$
  $a_k \leftarrow (1 + \sqrt{1 + 4a_{k-1}^2})/2$
  $\boldsymbol{w}^k \leftarrow \boldsymbol{v}_k + (a_{k-1} - 1)/a_k(\boldsymbol{v}_k - \boldsymbol{v}_{k-1})$
end while
Output: $\boldsymbol{w}^k$

In the top plot of Fig. 3.7, we compare SR3 and a primal-dual algorithm [84] on the objectives (3.25) and (3.24), respectively. Algorithm 5 converges as fast as the state-of-the-art method of [84]; 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 3.2 do not apply) and the equivalent of $\mathbf{F}_\kappa$, see (3.4), is still ill-conditioned. Nonetheless, since SR3 gives a primal-only method, it is straightforward to accelerate using FISTA [37]. In Fig. 3.7, 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 7.

We do not compare the support recovery of the two formulations, (3.24) and (3.25), 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 $\boldsymbol{x} = \text{vec}(\mathbf{X})$,

we have

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

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

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

where

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

and $\odot$ is element-wise multiplication. The SR3 formulation (3.25) 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 $\boldsymbol{w} = \operatorname{vec}\left(\circ\sqrt{\mathbf{W}_x^{②} + \mathbf{W}_y^{②}} \right)$, where $\circ\sqrt{A}$ and $A^{②}$ 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

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

$\mathbf{F}_\kappa^\top \mathbf{F}_\kappa$ 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 3.8, 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 simple to apply the FISTA algorithm to the reduced problem (3.5), improving the convergence rates.

3.3.3 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 matters. 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.

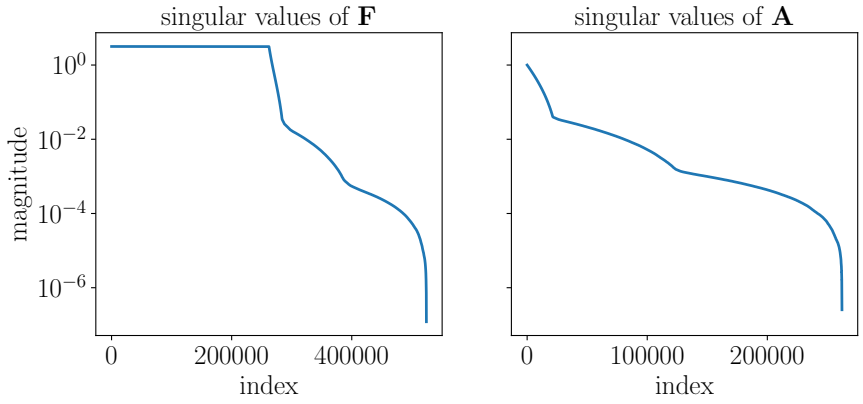


Figure 3.8: Singular values (ordered by magnitude) of $\mathbf{F}_\kappa$ (left panel) and $\mathbf{A}$ (right panel) in the TV example.

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 3.9 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 both 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 3.9, 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 3.9, 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

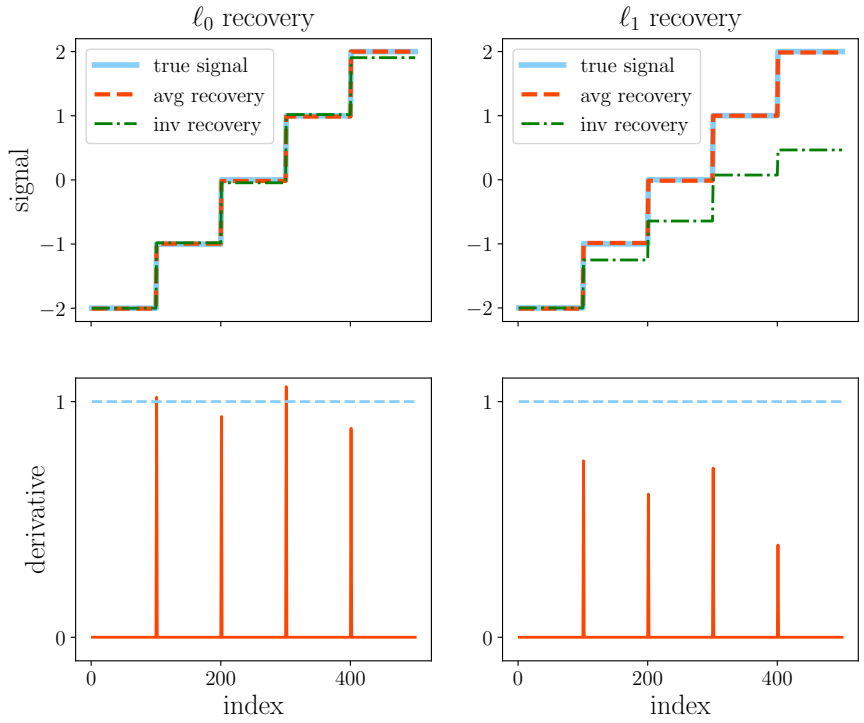


Figure 3.9: SR3 TV regularization result on synthetic data. The first row plots the averaging recovery signal (dashed red line), integrating recovery signal (dot dashed green line) and the true signal (solid blue line). Second row plots the discretized derivative (solid red line) and true magnitude (dashed blue line). First column contain the results come from ℓ_0 regularization, second column is from ℓ_1.

integration in this example) to reconstruct the signal given $\boldsymbol{w}$. From the second row of Figure 3.9 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.

3.3.4 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 [367], combinatorial optimization [290, 73], and seismic data interpolation/denoising [266, 14].

We compare classic rank penalty approaches using the nuclear norm (see e.g. [290]) 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 [346], reverse-

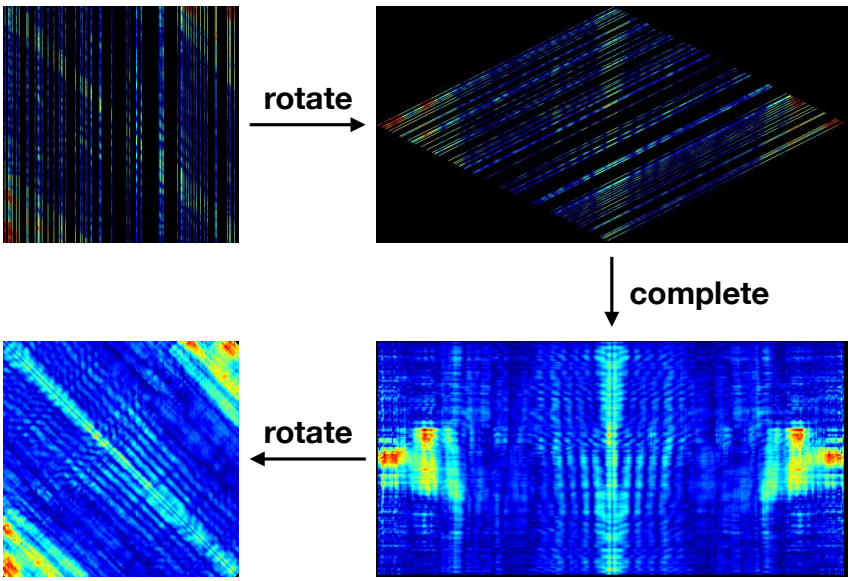


Figure 3.10: Interpolating a frequency slice from the Gulf of Suez dataset. Clockwise we see subsampled data in the source-receiver domain; transformation of the data to the midpont-offset domain, interpolation, and inverse transform back to the source/receiver domain.

time migration [34] and multiple removal methods [344]. 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, arranged as a matrix by source/receiver, see the left plot of Fig. 3.10. Fully sampled seismic data has a fast decay of singular values, while sub-sampling breaks this decay [14]. A convex formulation for matrix completion with nuclear norm is given by [290]

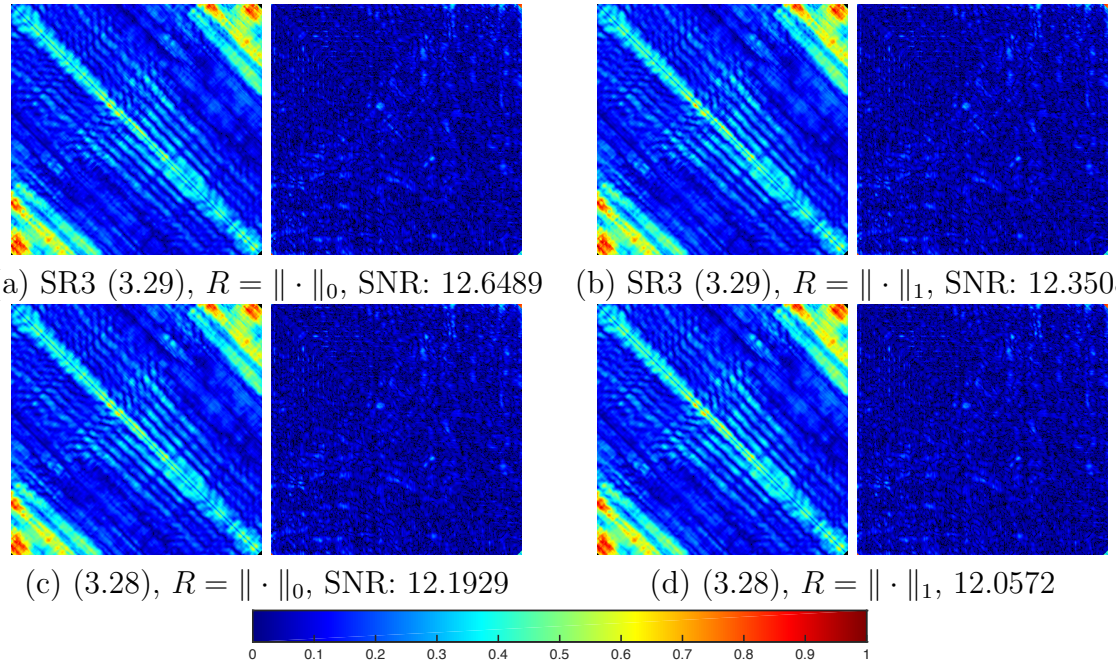
$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{X})) \tag{3.26}$$

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

The SR3 model relaxes (3.28) to obtain the formulation

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{W})) + \frac{\kappa}{2}\|\mathbf{W} - \mathbf{X}\|_F^2. \tag{3.27}$$

To find $\mathbf{X}(\mathbf{W})$, the minimizer of (3.29) with respect to $\mathbf{X}$, we solve a least squares problem. The $\mathbf{W}$ update requires thresholding the singular values of $\mathbf{X}(\mathbf{W})$.



(a) SR3 (3.29), $R = \|\cdot\|_0$, SNR: 12.6489 (b) SR3 (3.29), $R = \|\cdot\|_1$, SNR: 12.3508

(c) (3.28), $R = \|\cdot\|_0$, SNR: 12.1929 (d) (3.28), $R = \|\cdot\|_1$, 12.0572

Figure 3.11: 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.

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

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{X})) \tag{3.28}$$

and

$$\min_{\mathbf{X},\mathbf{W}} \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{D}\|_F^2 + \lambda R(\sigma(\mathbf{W})) + \frac{\kappa}{2}\|\mathbf{W} - \mathbf{X}\|_F^2 \,, \tag{3.29}$$

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 a support estimate, as in the compressive sensing examples.

In Figure 3.10, 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 [14], with $m = \frac{1}{2}(s+r)$ and $h = \frac{1}{2}(s-r)$, increasing the dimension to 401×801. We then solve (3.29) to interpolate the slice, and compare with the original to get a signal-to-noise ratio (SNR) of 9.7 (last panel in Fig. (3.10)). The SNR obtained by solving (3.28) is 9.2.

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 and when $R = \ell_0$, we range λ from 200 to 400.

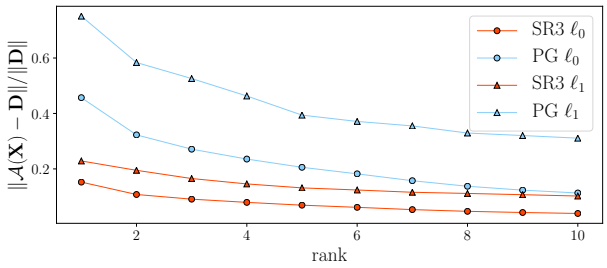


Figure 3.12: Pareto frontiers (best fit achievable for each rank) for (3.28) with $R = \ell_1, R = \ell_0$, and for corresponding SR3 formulations (3.29), describing the best fits of observed values achievable for a given rank (obtained across regularizers for the four formulations). ℓ_0 formulations are more efficient than those with ℓ_1, and SR3 formulations (3.29) are more efficient classic formulations (3.28).

Results. Results are shown in Figures 3.11 and 3.12. 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 (3.28), obtaining a maximum SNR of 12.6, see Figure 3.11.

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 3.12. The Pareto curves obtained using the SR3 approach are lower still, through the relaxation.

3.3.5 SR3 for Group Sparsity

Group sparsity is a composite sparse regularizer used in multi-task learning to regularize under-determined learning tasks by introducing redundancy in the solution vectors. Consider a set of under-determined linear systems,

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

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

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

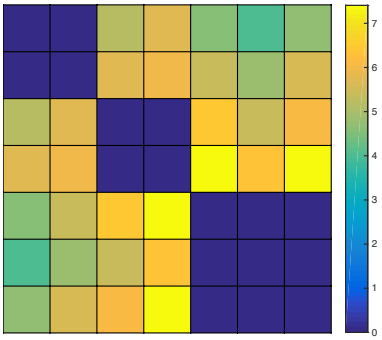


Figure 3.13: Pairwise distance between all decision variables of different tasks obtained by SR3.

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

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

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

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

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 3.13. 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).

3.4 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 also suggests broader applicability of SR3. Specially, we can also 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 (3.2) generalizes to

$$\min_{\boldsymbol{x},\boldsymbol{w}} f(\mathbf{A}, \boldsymbol{x}, \boldsymbol{b}) + \lambda R(\boldsymbol{w}) + \frac{\kappa}{2}\|\mathbf{C}\boldsymbol{x} - \boldsymbol{w}\|^2. \tag{3.30}$$

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.

3.5 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 $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ 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 $f\colon \mathbb{R}^n \to \overline{\mathbb{R}}$ 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

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

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 (3.1) and (3.5), which are defined below, for both convex and nonconvex regularizers $R(\cdot)$.

Definition 5 (Stationarity)**.** *We call* $\hat{\boldsymbol{x}}$ *the stationary point of* (3.1) *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* (3.5) *if,*

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

Moreau Envolope and Prox Operators

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

$$f_\eta(x) := \inf_z \left\{ f(z) + \tfrac{1}{2\eta}\|z - x\|^2 \right\}, \tag{3.31}$$

$$\text{prox}_{\eta f}(x) := \operatorname*{argmin}_z \left\{ \eta f(z) + \tfrac{1}{2}\|z - x\|^2 \right\}, \tag{3.32}$$

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 13 (Regularization properties of the envelope)**.** *Let $f \colon \mathbb{R}^n \to \mathbb{R}$ 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

$$0 \leq f(x) - f_\eta(x) \leq \frac{L^2\eta}{2} \qquad \textit{for all } x \in \mathbb{R}^n. \tag{3.33}$$

Proof. See Theorem 2.26 of [295]. □

However, when f is not convex, f_η may no longer be smooth as we show in Figure 3.14 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 [96].

- $f(\cdot) = \|\cdot\|_1$. 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.

  Lemma 4 (ℓ_1)**.** *The prox operator of ℓ_1 is an element-wise soft-thresholding action on the given vector.*

$$\boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_1 \quad \Rightarrow$$

$$x_i = \begin{cases} y_i - \eta, & y_i > \eta \\ 0, & |y_i| \leq \eta \\ y_i + \eta, & y_i < -\eta \end{cases}. \tag{3.34}$$

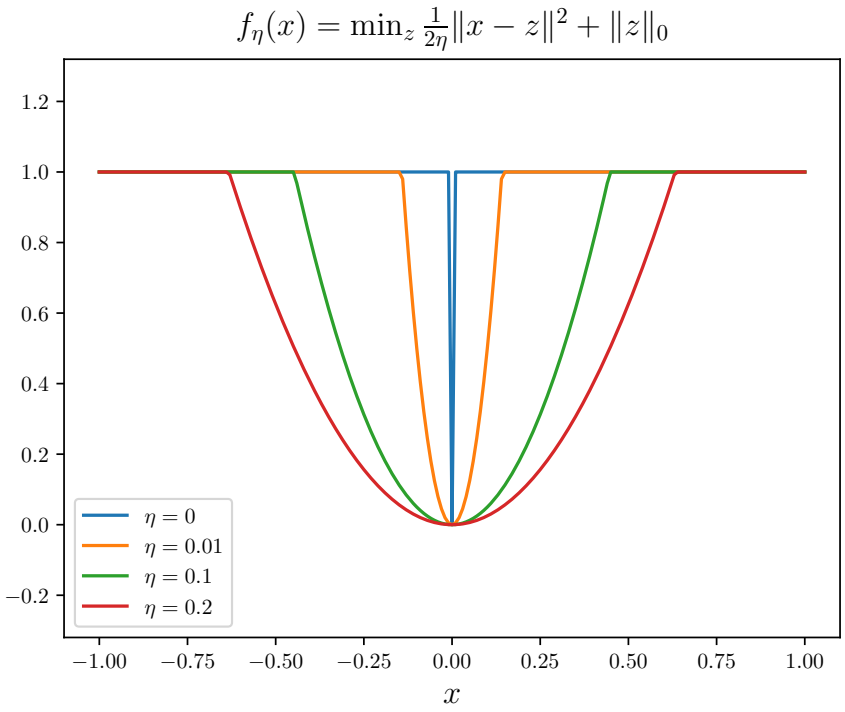


Figure 3.14: Envelope functions indexed by the parameter η, for $f = \|\cdot\|_0$. In contrast to the convex case, here all f_η are nonsmooth and nonconvex.

Proof. Note that the optimization problem may be written as

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{x}} \ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_1 \\ &\qquad = \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{1}{2}\sum_{i=1}^{n}(x_i - y_i)^2 + \eta|x_i| \ , \end{aligned} \tag{3.35}$$

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

$$\begin{aligned} &\partial_{x_i}\left(\frac{1}{2}(x_i - y_i)^2 + \eta|x_i|\right) \\ &= \begin{cases} x_i - y_i + \eta, & x_i > 0 \\ x_i - y_i + \{z : |z| \le \eta\}, & x_i = 0 \\ x_i - y_i - \eta, & x_i < 0 \end{cases} . \end{aligned} \tag{3.36}$$

After checking the possible stationary points given this formula for the subdifferential, it is simple to derive (3.34). □

- $f(\cdot) = \|\cdot\|_0$. 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 5 (ℓ_0)**.** *The prox operator of ℓ_0 is simple, element-wise hard-thresholding:*

$$\begin{aligned} \boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\text{argmin}} \ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\|_0 \quad \Rightarrow \\ x_i = \begin{cases} y_i, & |y_i| > \sqrt{2\eta} \\ 0, & |y_i| \leq \sqrt{2\eta} \end{cases}. \end{aligned} \tag{3.37}$$

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

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

  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 (3.37) 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 6 (ℓ_2^2)**.** *The prox of ℓ_2^2 is scaling.*

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

  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 7.

$$\begin{aligned} \boldsymbol{x} = \text{prox}_{\eta f}(\boldsymbol{y}) = \underset{\boldsymbol{x}}{\text{argmin}} \ \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 + \eta\|\boldsymbol{x}\| \quad \Rightarrow \\ \boldsymbol{x} = \begin{cases} \frac{\|\boldsymbol{y}\|-\eta}{\|\boldsymbol{y}\|}\boldsymbol{y}, & \|\boldsymbol{y}\| > \eta \\ \boldsymbol{0}, & \|\boldsymbol{y}\| \leq \eta \end{cases}. \end{aligned}$$

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

$$\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2+\eta\|\boldsymbol{x}\| \tag{3.38}$$

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

Algorithm 8 Proximal gradient descent

1: **Input:** $\boldsymbol{x}_0, \eta$
2: **Initialize:** $k = 0$
3: **while** not converged **do**
4: $\quad k \leftarrow k+1$
5: $\quad \boldsymbol{x}_k \leftarrow \text{prox}_{\eta g}(\boldsymbol{x}_{k-1} - \eta\nabla f(\boldsymbol{x}_{k-1}))$
6: **end while**
7: **Output:** $\boldsymbol{x}_k$

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 6 [96]. This algorithm has been studied extensively. Among other results, we have

Theorem 14 (Proximal Gradient Descent)**.** *Assume $p = f + g$ and both p and g are closed convex functions. Let p^* 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. [37, 96, 275] and the tutorial section 4.4 of [11].

Theoretical Results

In the main text, it is demonstrated that SR3 (3.5) outperforms the standard regression problem (3.1), 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 $\mathbf{F}_\kappa$

Proof of Theorem 9 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,

$$\begin{aligned}\mathbf{F}_\kappa^\top \mathbf{F}_\kappa &= \kappa \mathbf{I} - \kappa^2 \mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top \\ \sigma_i(\mathbf{F}_\kappa^\top \mathbf{F}_\kappa) &= \kappa - \kappa^2 \sigma_{n-i+1}(\mathbf{C}\mathbf{H}_\kappa^{-1}\mathbf{C}^\top).\end{aligned} \tag{3.39}$$

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 $\mathbf{C} = \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^\top$. When $n \geq d$ and $\mathbf{C}$ is full rank, we know $\mathbf{\Sigma}_c$ is invertible and $\mathbf{V}_c$ is orthogonal. Then

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

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

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

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) $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$, and $\mathbf{V} \in \mathbb{R}^{m\times m}$. We have

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

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

$$\mathbf{\Sigma} = \begin{bmatrix}\hat{\mathbf{\Sigma}} \\ \mathbf{0}\end{bmatrix}, \quad \mathbf{F}_\kappa^\top \mathbf{F}_\kappa = \mathbf{V}\hat{\mathbf{\Sigma}}^\top(\mathbf{I} + \hat{\mathbf{\Sigma}}^2/\kappa)^{-1}\hat{\mathbf{\Sigma}}\mathbf{V}^\top \tag{3.40}$$

And when $m < n$,

$$\mathbf{\Sigma} = \begin{bmatrix}\hat{\mathbf{\Sigma}} & \mathbf{0}\end{bmatrix}, \quad \mathbf{F}_\kappa^\top \mathbf{F}_\kappa = \mathbf{V}\begin{bmatrix}\hat{\mathbf{\Sigma}}^\top(\mathbf{I} + \hat{\mathbf{\Sigma}}^2/\kappa)^{-1}\hat{\mathbf{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{V}^\top \tag{3.41}$$

(3.8) and (3.9) 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 (3.9), we have

$$\begin{aligned}\sigma_{\max}(\mathbf{F}_\kappa) &= \frac{\sigma_{\max}(\mathbf{A})}{\sqrt{1 + \sigma_{\max}(\mathbf{A})^2/\kappa}} \quad \text{and} \\ \sigma_{\min}(\mathbf{F}_\kappa) &= \frac{\sigma_{\min}(\mathbf{A})}{\sqrt{1 + \sigma_{\min}(\mathbf{A})^2/\kappa}}\,.\end{aligned}$$

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

Proof of Theorem 10. For the iterates of the proximal gradient method, we have

$$\boldsymbol{x}_{k+1} = \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \eta\nabla f(\boldsymbol{x}_k))\|^2 + \eta g(\boldsymbol{x})$$

and from the first order optimality condition we have

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

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

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

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.

Proof of Theorem 11. The result is immediate from combining Theorem 10 and Theorem 9.

Proof of Corollary 2. The result is immediate from combining Theorem 10 and Corollary 1.

Characterizing Optimal Solutions of SR3

In this section, we quantify the relation between the solution of (3.1) and (3.5) when $\mathbf{C} = \mathbf{I}$. In this analysis, we fix κ as a constant and set $\mathbf{C} = \mathbf{I}$.

Lemma 8 (Optimality conditions for (3.1) and (3.5))**.** *Define the sets*

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

where $\mathbf{H}_\kappa = \mathbf{A}^\top \mathbf{A} + \kappa \mathbf{I}$*, as above. These sets contain the subgradients of* (3.1) *and* (3.5)*. If we assume* $\hat{\boldsymbol{x}}$ *and* $\hat{\boldsymbol{w}}$ *are the (stationary) solutions of* (3.1) *and* (3.5)*, 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

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

Proof. As $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{w}}$ are the (stationary) solutions of (3.1) and (3.5), we have

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

Then,

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

□

Proof of Theorem 12 Using the definitions of Lemma 8, we have

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

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 12 provides some explicit bounds on the distance between these solutions.

Corollary 3. *If* $\mathbf{A}^\top \mathbf{A} = \mathbf{I}$, *then* $\operatorname{dist}(\mathbf{0}, \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)) = 0$, *i.e.* $\hat{\boldsymbol{w}}$ *is the stationary point of* (3.1). *If* $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$, *then* $\operatorname{dist}(\mathbf{0}, \mathcal{S}_1(\hat{\boldsymbol{w}}, \lambda_1)) \leq 1/(1 + 2\kappa)$.

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 12 then implies the result. □

Implementation of ℓ_q proximal operator.

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

$$\begin{aligned} f'_{\alpha,p}(x; z) &= \frac{1}{\alpha}(x - |z|) + p x^{p-1}, \\ f''_{\alpha,p}(x; z) &= \frac{1}{\alpha} + p(p-1)x^{p-2}. \end{aligned} \tag{3.42}$$

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 $0 \leq x < \tilde{x}$, and $f''_{\alpha,p}(x; z) > 0$ when $x > \tilde{x}$.

- If $f'_{\alpha,p}(\tilde{x}; z) \geq 0$, we have $f'_{\alpha,p}(x; z) \geq 0$, for all $x \geq 0$. Then $\operatorname{argmin}_{x \geq 0}\ f_{\alpha,p}(x; z) = 0$.
- 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

$$\operatorname*{argmin}_{x \geq 0}\ f_{\alpha,p}(x; z) = \begin{cases} 0, & f_{\alpha,p}(0; z) \leq f_{\alpha,p}(\bar{x}; z) \\ \bar{x}, & f_{\alpha,p}(0; z) > f_{\alpha,p}(\bar{x}; z) \end{cases}.$$

Chapter 4

ROBUST AND SCALABLE METHODS FOR THE DYNAMIC MODE DECOMPOSITION

The dynamic mode decomposition (DMD) is a broadly applicable dimensionality reduction algorithm that approximates a matrix containing time-series data by the outer product of a matrix of exponentials, representing Fourier-like time dynamics, and a matrix of coefficients, representing spatial structures. This interpretable spatio-temporal decomposition is commonly computed using linear algebraic techniques in its simplest formulation or a nonlinear optimization procedure within the variable projection framework. For data with sparse outliers or data which are not well-represented by exponentials in time, the standard Frobenius norm fit of the data creates significant biases in the recovered time dynamics. As a result, practitioners are left to clean such defects from the data manually or to use a black-box cleaning approach like robust PCA. As an alternative, we propose a framework and a set of algorithms for incorporating robust features into the nonlinear optimization used to compute the DMD itself. The algorithms presented are flexible, allowing for regularizers and constraints on the optimization, and scalable, using a stochastic approach to decrease the computational cost for data in high dimensional space. Both synthetic and real data examples are provided.

4.1 Introduction

Dimensionality reduction is a critically enabling tool in machine learning applications. Specifically, extracting the dominant low-rank features from a high-dimensional data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ allows one to efficiently perform tasks associated with clustering, classification and prediction. As defined by [103], *linear* dimensionality reduction methods solve an optimization problem with objective $f_{\mathbf{X}}(\cdot)$ over a manifold $\mathcal{M}$ to produce a linear transformation P which maps the columns of $\mathbf{X}$ to a lower dimensional space. Many popular methods can be written in this framework by an appropriate definition of $f_{\mathbf{X}}(\cdot)$ and specification of the manifold $\mathcal{M}$. For instance, the principal component analysis (PCA) may be written as

$$\hat{M} = \underset{M\in\mathcal{M}}{\operatorname{argmin}} \|\mathbf{X} - MM^{\top}\mathbf{X}\|_F\,, \qquad \mathcal{M} = \mathcal{O}^{m\times k}\,, \tag{4.1}$$

where $\mathcal{O}^{m\times k}$ is the manifold of $m \times k$ matrices with orthonormal columns, i.e. $\mathcal{M}$ is a Stiefel manifold. The map P is then given by $\hat{M}^{\top}$. One of the primary conclusions of the survey [103], is that — aside from the PCA itself — many of the common methods for linear

dimensionality reduction based on eigenvalue solvers are actually sub-optimal heuristics and the direct solution of the optimization problem (4.1) should be preferred.

In this manuscript, we consider a particular linear dimensionality reduction technique: the dynamic mode decomposition (DMD). In the past decade, the DMD has been applied to the analysis of fluid flow experiments and simulations, machine learning enabled control systems, and Koopman spectral analysis, among other data-intensive problems described by dynamical systems. The success of the algorithm is largely due to the interpretability of the low-rank spatio-temporal modes it generates in approximating the dominant features of the data matrix $\mathbf{X}$. The DMD was originally defined to be the output of an algorithm for characterizing time-series measurements of fluid flow data [305, 306]. It was later reformulated by [337] as a least-squares regression problem whereby the DMD could be stably computed using a Moore-Penrose pseudo-inverse and an eigenvalue decomposition. An earlier though less commonly used formulation, the *optimized* DMD [90], can be phrased as the optimization problem

$$\hat{M} = \underset{M \in \mathcal{M}}{\operatorname{argmin}} \|\mathbf{X} - MM^{\dagger}\mathbf{X}\|_F \ , \quad \mathcal{M} = \mathbf{\Phi}(\mathbb{C}^k) \ , \tag{4.2}$$

where the map $\boldsymbol{\alpha} \mapsto \mathbf{\Phi}(\boldsymbol{\alpha})$ defines a matrix with columns corresponding to exponential time dynamics (see Section 4.2.1) and $M^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of M. This can be thought of as a best-fit linear dynamical system approximation of the data. In agreement with the conclusions of [103], the optimized DMD, while more costly to compute, is more robust to additive noise than the exact DMD and its noise-corrected alternatives [109, 20]. It is also more flexible than the exact DMD, allowing for non-equispaced snapshots. While the optimized DMD does not fit directly into the optimization framework of [103], which is defined for $\mathcal{M}$ either a Stiefel manifold or a Grassmannian manifold, it can be computed efficiently using classical variable projection methods [163, 20].

The DMD has been used in a variety of fields where the nature of the data can lead to corrupt and noisy measurements. This includes applications ranging from neuroscience [58] to video processing [172, 137] to fluid dynamics [305, 306, 175, 109]. Although the Frobenius norm used in the definition of the optimized DMD (4.2) is appealing due to its physical interpretability in many applications (energy, mass, etc.), it has significant flaws that can severely limit its applicability. Specifically, corrupt data or large noise fluctuations can lead to significant deformation of the DMD approximation of the data because the Frobenius norm implicitly assigns a very low probability to such outliers (see Section 4.2.2). In practice, these outliers are often removed from the data manually or using a black-box filtering approach like robust PCA [228, 358, 79]. However, such approaches ignore the structure of the DMD approximation and may introduce biases of their own. Further, it is desirable that DMD methods not only be robust to "noisy" outliers but also to non-exponential structure in the data. We therefore develop an alternative approach to increase the robustness of the DMD. In particular, we modify the optimized DMD definition (4.2) to incorporate ideas from the field of robust statistics [245, 195] in order to produce a decomposition that is significantly less sensitive to outliers in the data.

Because the new problem formulation incorporates robust norms, many of the efficient

strategies used in variable projection algorithms for problems defined in the Frobenius norm are no longer available. To remedy this, we develop a number of algorithms based on modern variable projection methods [18, 15] which exploit the structure of the DMD for increased performance. In particular, we can incorporate nonsmooth features, such as regularizers and constraints, and scale to large problems using stochastic variance reduction techniques.

This flexible architecture allows us to impose physically relevant constraints on the optimization that are critical for tasks such as future-state prediction. For instance, we can impose the constraint that the real parts of the DMD eigenvalues are non-positive, thus ensuring that solutions do not grow to infinity when forecasting.

The effect of noise on the DMD is a well-studied area. Controlling for the bias of the exact DMD in the presence of additive noise was treated by [185] and [109]. A Bayesian formulation of the DMD was presented by [326]. This formulation is flexible enough to incorporate robust statistics but this was not a focus of that work. [114] presented a robust formulation of exact DMD type, which complements the current work.

The rest of this manuscript is organized as follows. In Section 4.2, we provide some necessary preliminaries from the DMD, robust statistics, and variable projection literature and we present our problem formulation. A detailed description of the algorithms we use to solve the robust DMD formulation follows in Section 4.3. We apply these methods to synthetic data in Section 4.4. Finally, we provide some concluding remarks and describe possible future directions in Section 4.5.

4.2 Preliminaries

In this section, we outline some of the precursors of this work and present our problem formulation.

4.2.1 Dynamic mode decomposition

As mentioned above, the dynamic mode decomposition (DMD) corresponds to a best-fit linear dynamical model of the data. Because linear dynamics produce exponential functions in time, the DMD may be written as an exponential fitting problem. Let $\mathbf{X} \in \mathbb{C}^{m\times n}$ be a snapshot matrix whose rows $\boldsymbol{x}_j$ are samples of an n dimensional dynamical system at a set of m sample times t_j. For a given rank k, let $\boldsymbol{\alpha} \in \mathbb{C}^k$ be a vector of complex numbers specifying time dynamics. We then define the matrix $\boldsymbol{\Phi}(\boldsymbol{\alpha}; \boldsymbol{t})$ by

$$\Phi_{ij}(\boldsymbol{\alpha}) = e^{\alpha_j t_i} \ . \tag{4.3}$$

When it is clear in context, we often drop the dependence of $\boldsymbol{\Phi}$ on $\boldsymbol{\alpha}$ and $\boldsymbol{t}$.

Let $\mathbf{B} \in \mathbb{C}^{k\times n}$ be a matrix of coefficients for the exponential fit. The so-called *optimized DMD* [90] is defined to be the solution of the following optimization problem:

$$\min_{\boldsymbol{\alpha},\mathbf{B}} \frac{1}{2}\|\mathbf{X} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B}\|_F^2 \ . \tag{4.4}$$

The problem (4.4) is a large, nonlinear least squares problem; in particular it is highly non-convex. The classical variable projection framework provides an efficient method for computing a (local) solution.

Let

$$f_{\text{opt}}(\boldsymbol{\alpha}, \mathbf{B}) = \frac{1}{2}\|\mathbf{X} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B}\|_F^2 \, .$$

The classical variable projection framework is based on the observation that for a fixed $\boldsymbol{\alpha}$, it is easy to optimize f_{opt} in $\mathbf{B}$. In fact, for the least squares case, we have a closed form expression

$$\mathbf{B}(\boldsymbol{\alpha}) := \arg\min_{\mathbf{B}} f_{\text{opt}}(\boldsymbol{\alpha}, \mathbf{B}) = \boldsymbol{\Phi}(\boldsymbol{\alpha})^{\dagger}\mathbf{X}, \tag{4.5}$$

where $\boldsymbol{\Phi}(\boldsymbol{\alpha})^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of $\boldsymbol{\Phi}(\boldsymbol{\alpha})$. Let

$$\tilde{f}_{\text{opt}}(\boldsymbol{\alpha}) = \min_{\mathbf{B}} f_{\text{opt}}(\boldsymbol{\alpha}, \mathbf{B}) := \frac{1}{2}\|\mathbf{X} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B}(\boldsymbol{\alpha})\|_F^2.$$

The variable projection (VP) technique finds the minimizer of $\tilde{f}_{\text{opt}}(\boldsymbol{\alpha})$ using an iterative method. First and second derivatives of $\tilde{f}$ with respect to $\boldsymbol{\alpha}$ are easily computed [39]:

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} \tilde{f}_{\text{opt}}(\boldsymbol{\alpha}) &= \partial_{\boldsymbol{\alpha}} f_{\text{opt}}|_{\boldsymbol{\alpha}, \mathbf{B}(\boldsymbol{\alpha})} \\ \nabla^2_{\boldsymbol{\alpha}} \tilde{f}_{\text{opt}}(\boldsymbol{\alpha}) &= \left[\partial^2_{\boldsymbol{\alpha}} f_{\text{opt}} - \partial_{\boldsymbol{\alpha},\mathbf{B}} f_{\text{opt}} (\partial^2_{\mathbf{B}} f_{\text{opt}})^{-1} \partial_{\mathbf{B},\boldsymbol{\alpha}} f_{\text{opt}}\right]\Big|_{\boldsymbol{\alpha}, \mathbf{B}(\boldsymbol{\alpha})} . \end{aligned} \tag{4.6}$$

These formulas allow first- and second-order methods to be directly applied to $\tilde{f}_{\text{opt}}$, including steepest descent, BFGS, and Newton's method. The matrix $\mathbf{B}(\boldsymbol{\alpha})$ is updated every time $\boldsymbol{\alpha}$ changes. Gauss-Newton and Levenberg-Marquardt (LM) have been classically used for exponential fitting; these methods do not use the Hessian in (4.6), opting for simpler approximations. The method was used for exponential fitting by [163]. While VP originally referred to least-squares projection (using the closed-form solution $\mathbf{B}(\boldsymbol{\alpha})$ in (4.5)), follow-up work considered more general loss functions, using the term *projection* to refer to partial minimization [18, 15].

For practitioners, the optimized DMD may be less familiar than exact DMD [337]. We favor the optimized DMD for its performance on data with additive noise [20] and its flexibility. In particular, the optimized formulation enables the contributions of the current work. For a review of the DMD and its applications, see [337] and [210].

4.2.2 Robust Formulations

The optimized DMD problem (4.4) is formulated using the least-squares error norm, which is equivalent to assuming a Gaussian model on the errors between predicted and observed data:

$$\mathbf{X} = \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma^2 I).$$

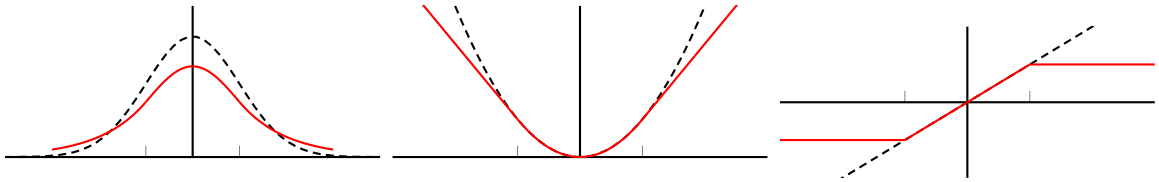


Figure 4.1: Gaussian (black dash) and Huber (red solid) Densities, Negative Log Likelihoods, and Influence Functions.

This error model, and the corresponding formulation, are vulnerable to outliers in the data. Both DMD and optimized DMD are known to be sensitive to outliers, so in practice data are 'pre-cleaned' before applying these approaches.

In many domains, formulations based on robust statics have become the method of choice for dealing with contaminated data. Two common approaches are

- to replace the LS penalty with one that penalizes deviations less harshly and
- to solve an extended problem that explicitly identifies outliers while fitting the model.

The first approach, often called M-estimation [195, 245], is illustrated in Figure 4.1. Replacing the least squares penalty by the Huber penalty

$$\rho(z) = \begin{cases} \frac{1}{2}|z|^2 & \text{if } |z| \leq \kappa \\ \kappa|z| - \frac{1}{2}\kappa^2 & \text{if } |r| > \kappa \end{cases}$$

corresponds to choosing the solid red penalty rather than the dotted black least squares penalty in the center panel of Figure 4.1. This corresponds to modeling errors $\boldsymbol{\epsilon}$ using the density $\exp(-\rho)$, which has heavier tails than the Gaussian (see left panel of Figure 4.1). Heavier tails means deviations (i.e. larger residuals) are more likely than under the Gaussian model, and so observations that deviate from the norm have less *influence*, i.e. effect on the fitted parameters $(\boldsymbol{\alpha}, \mathbf{B})$ than under the Gaussian model (see right panel of Figure 4.1). The M-estimator-DMD problem can be written as

$$\min_{\boldsymbol{\alpha},\mathbf{B}} \sum_{j=1}^{n} \rho(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B}_j) := \sum_{j=1}^{n} \rho_j(\boldsymbol{\alpha}, \mathbf{B}),$$

where the sum is run across columns.

Another approach, called trimmed estimation, builds on M-estimation by coupling explicit outlier identification/removal with model fitting. The trimmed DMD formulation for any penalty ρ is given by

$$\min_{\boldsymbol{\alpha},\mathbf{B}} \sum_{l=1}^{h} \rho_{j_l}(\boldsymbol{\alpha}, \mathbf{B}), \tag{4.7}$$

where $\rho_{j_1}(\boldsymbol{\alpha}, \mathbf{B}) \leq \cdots \leq \rho_{j_h}(\boldsymbol{\alpha}, \mathbf{B})$ are the first h order statistics of the objective values and $\{j_1, \ldots, j_h\} \subseteq \{1, \ldots, n\}$. Interpreting the loss ρ_j as the negative log likelihood of the jth observed column, it is clear that trimming jointly fits a likelihood model while simultaneously eliminating the influence of all low-likelihood observations. An equivalent formulation to (4.7) replaces the order statistics with explicit weights

$$\min_{\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}} \sum_{j=1}^{n} w_j \rho_j(\boldsymbol{\alpha}, \mathbf{B}), \quad 0 \leq w_j \leq 1, \quad \mathbf{1}^\top \boldsymbol{w} = h. \tag{4.8}$$

The reader should verify that (4.8) and (4.7) are equivalent.

Trimmed M-estimators were initially introduced by [301] in the context of least-squares regression. The author's original motivation was to develop linear regression estimators that have a high breakdown point (in this case 50%) and good statistical efficiency (in this case $n^{-1/2}$)[1]. For a number of years, the difficulty of efficiently optimizing LTS problems limited their application. However, recent work has made it possible to efficiently apply trimming to general models [364, 12]. We show how to incorporate trimming into the robust DMD framework.

4.2.3 Regularization

Optimized DMD allows prior knowledge to be incorporated into the optimization formulation, either through constraints on variables, or regularization terms. In all exponential fitting problems, the real parts of $\boldsymbol{\alpha}$ coefficients play a major role in explaining the data because of the exponential growth of $\boldsymbol{\Phi}(\boldsymbol{\alpha})$. A natural regularization is to restrict the magnitudes of the real parts of $\boldsymbol{\alpha}$, imposing the constraint $\text{real}(\boldsymbol{\alpha}) \leq \gamma$ with γ chosen by the user. We write the constraint as follows:

$$r(\boldsymbol{\alpha}) = \begin{cases} 0 & \text{if real}(\boldsymbol{\alpha}) \leq \gamma \\ \infty & \text{if real}(\boldsymbol{\alpha}) > \gamma. \end{cases}$$

This is a simple convex function that admits a trivial proximal operator (see [96]): the projection onto the shifted left half-plane in $\mathbb{C}^k$. The VP technique can be easily adapted to incorporate such functions on $\boldsymbol{\alpha}$.

Constraints and penalties can also be imposed on the matrix $\mathbf{B}$. We assume that only smooth separable regularization penalties can be used; and in this case, the regularization is added to the g function.

4.2.4 Problem formulation

Let $h(\mathbf{B})$ and $r(\boldsymbol{\alpha})$ be convex regularization terms. We formulate the general robust DMD problem as follows:

---

[1]Breakdown refers to the percentage of outlying points which can be added to a dataset before the resulting M-estimator can change in an unbounded way.

$$\min_{\boldsymbol{\alpha},\mathbf{B},\boldsymbol{w}} f(\boldsymbol{\alpha},\mathbf{B},\boldsymbol{w}) := g(\boldsymbol{\alpha},\mathbf{B},\boldsymbol{w}) + r(\boldsymbol{\alpha}) + s(\boldsymbol{w}) \ , \tag{4.9}$$

where $r(\boldsymbol{\alpha})$ encodes optional regularization functions (or constraints) for $\boldsymbol{\alpha}$ (see Section 4.2.3) and

$$g(\boldsymbol{\alpha},\mathbf{B},\boldsymbol{w}) = \sum_{j=1}^{n} w_j \rho\left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B}_{\cdot j}\right) + q(\mathbf{B}_{\cdot j}) \tag{4.10}$$

with ρ any differentiable penalty, $q(\mathbf{B}_{\cdot j})$ representing potential regularizer for columns of $\mathbf{B}$, and $s(\boldsymbol{w})$ encoding the capped simplex constraints:

$$s(\boldsymbol{w}) = \begin{cases} 0 & \text{if } 0 \le w_j \le 1,\ \mathbf{1}^\top \boldsymbol{w} = h \\ \infty & \text{else.} \end{cases} \tag{4.11}$$

These constraints are explained in Section 4.2.2. The $\boldsymbol{w}$ variables select the best-fit h columns of the data, and only use those values to update $\boldsymbol{\alpha}$. Since each $w_j \in [0,1]$ rather than $\{0,1\}$, the solutions do not have to be integral. However, for any fixed $(\mathbf{B},\boldsymbol{\alpha})$ there exists a vertex solution, since the subproblem in $\boldsymbol{w}$ with the other variables fixed is a linear program. The function $s(\boldsymbol{w})$ admits a simple proximal operator, which is the projection onto the intersection of the h-simplex with the unit cube[2].

Setting $h = n$ forces $w_j = 1$ for each column, eliminating trimming completely, and reducing (4.9) to a simpler regularized M-estimation form of DMD.

For notational convenience, we define a matrix-valued penalty function

$$\boldsymbol{\rho}(A) := \begin{bmatrix} \rho(A_{1,1}) & \cdots & \rho(A_{1,n}) \\ \vdots & \ddots & \vdots \\ \rho(A_{m,1}) & \cdots & \rho(A_{m,n}) \end{bmatrix}.$$

In this notation, we can write

$$g(\boldsymbol{\alpha},\mathbf{B},\boldsymbol{w}) = \mathbf{1}^\top \boldsymbol{\rho}(\mathbf{X} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\mathbf{B})\boldsymbol{w} + q(\mathbf{B}),$$

which makes derivative computations straightforward.

Our numerical examples use constraints for $\boldsymbol{\alpha}$, but do not regularize $\mathbf{B}$, that is, $q(\mathbf{B}) \equiv 0$. However, we consider separable penalties q in the algorithmic description to preserve the generality of (4.9).

4.2.5 Gradient computations

We need to compute the gradient of the penalty function (4.10) with respect to the entries of $\boldsymbol{\alpha}$ and $\mathbf{B}$. In all methods, we treat treat the real and imaginary components of α_j and B_{ji} as independent, real-valued parameters.

---

[2]This set is called the *capped simplex*, and admits fast projections [12].

Consider a complex number $z = x + \boldsymbol{i}y$. We write derivative formulas in the Wirtinger sense, computing partial derivatives with respect to the complex variables. The derivatives for the real components can then be recovered from the formulas

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - \boldsymbol{i}\frac{\partial}{\partial y}\right). \tag{4.12}$$

Let $g(z)$ be a function of z which can be written as $g(z) = G(z, \bar{z})$ where G is differentiable with respect to both z and $\bar{z}$. The Wirtinger derivative of g is then the partial derivative of G with respect to z, treating $\bar{z}$ as a constant. For example, the Huber penalty may be written as

$$\rho(z) = H(z, \bar{z}; \kappa) = \begin{cases} \kappa\sqrt{z\bar{z}} - \frac{1}{2}\kappa^2, & |z| \geq \kappa \\ \frac{1}{2}z\bar{z}, & |z| < \kappa \end{cases}.$$

The Wirtinger derivative of the Huber penalty is then

$$\rho'(z) = \frac{\partial}{\partial z}H(z, \bar{z}; \kappa) = \begin{cases} \dfrac{\kappa\bar{z}}{2\sqrt{z\bar{z}}}, & |z| < \kappa \\ \frac{1}{2}\bar{z}, & |z| \geq \kappa \end{cases}.$$

The gradients of f with respect to $\boldsymbol{\alpha}$ and $\mathbf{B}$ can then be computed using the chain rule:

$$\begin{aligned} \nabla_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}) &= -\operatorname{diag}\left[\mathbf{B}\operatorname{Diag}(\boldsymbol{w})\boldsymbol{\rho}'(\mathbf{X} - \boldsymbol{\Phi}\mathbf{B})^\top(\operatorname{Diag}(\boldsymbol{t})\boldsymbol{\Phi})\right] \\ \nabla_{\mathbf{B}} g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}) &= -\boldsymbol{\Phi}^\top\boldsymbol{\rho}'(\mathbf{X} - \boldsymbol{\Phi}\mathbf{B})\operatorname{Diag}(\boldsymbol{w}) + \mathbf{B}^\top\nabla q(\mathbf{B}) \\ \nabla_{\boldsymbol{w}} g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}) &= \boldsymbol{\rho}(\mathbf{X} - \boldsymbol{\Phi}\mathbf{B})^\top\mathbf{1}, \end{aligned} \tag{4.13}$$

where we define

$$\operatorname{diag}(A) := \begin{bmatrix} a_{11} \\ \vdots \\ \vdots \\ a_{nn} \end{bmatrix}, \qquad \operatorname{Diag}(a) := \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_n \end{bmatrix}.$$

4.3 Methods

In this section, we develop numerical approaches for (4.9).

4.3.1 Variable projection framework

To compute the robust optimized DMD, we apply the variable projection (VP) technique to the optimization problem (4.9). Define the reduced function $\tilde{f}$ and implicit solution $\mathbf{B}(\boldsymbol{\alpha})$ by

$$\begin{aligned} \tilde{f}(\boldsymbol{\alpha}, \boldsymbol{w}) &= \min_{\mathbf{B}} f(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}), \\ \mathbf{B}(\boldsymbol{\alpha}, \boldsymbol{w}) &= \operatorname*{argmin}_{\mathbf{B}} f(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}), \end{aligned} \tag{4.14}$$

where f is as defined in (4.9). The gradient formula (4.6) holds for a very broad problem class. In particular, it holds as long as the following conditions are satisfied [294, Theorem 10.58]:

1. $g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w})$ is level-bounded in $\mathbf{B}$ locally uniformly in $\boldsymbol{\alpha}$; in particular for any compact subset of $\boldsymbol{\alpha}$, the union of sublevel sets $\{\mathbf{B} : g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w}) \leq \gamma\}$ is bounded.

2. The gradient of $g(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w})$ exists and is continuous for all $(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w})$.

3. $\mathbf{B}(\boldsymbol{\alpha}, \boldsymbol{w})$ is unique.

Several assumptions on g, $\boldsymbol{\Phi}$, and q can be made to ensure these conditions hold. For example, if g is differentiable, convex, coercive[3] in $\mathbf{B}$, and $\boldsymbol{\Phi}(\boldsymbol{\alpha})$ has full rank, then the result holds. If the same conditions hold for g, $\boldsymbol{\Phi}(\boldsymbol{\alpha})$ does not have full rank, but q is strictly convex, the result holds as well. The derivative formulas are valid for all of the examples in the paper, and we have

$$\nabla \tilde{f}(\boldsymbol{\alpha}, \boldsymbol{w}) = \partial_{\boldsymbol{\alpha}, \boldsymbol{w}} f(\boldsymbol{\alpha}, \mathbf{B}, \boldsymbol{w})|_{\boldsymbol{\alpha}, \mathbf{B}(\boldsymbol{\alpha}), \boldsymbol{w}} \,. \tag{4.15}$$

We refer to partially minimizing over f over $\mathbf{B}$ as the *inner problem* and minimizing $\tilde{f}$ as the *outer problem*. When f is convex and smooth with respect to $\mathbf{B}$, a lot of fast optimization algorithms can be applied to the inner problem. The inner problem is also embarrassingly parallelizable, and we make use of the problem structure in algorithm design. The general VP strategy is to use an iterative method to compute a (local) minimizer of the reduced function (4.14).

Solving (4.9) requires optimization procedures for both the inner and outer problems. We discuss these algorithms in the next two subsections.

4.3.2 Batch methods

Consider the problem class where g in (4.9) is convex and continuously differentiable with respect to $\mathbf{B}$. In this case, the inner problem decouples into n independent subproblems:

$$\boldsymbol{b}_j(\boldsymbol{\alpha}, \boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{b}} \quad w_j \rho \left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha}) \boldsymbol{b} \right) + q(\boldsymbol{b}), \quad j = 1, \ldots, n. \tag{4.16}$$

We use BFGS to solve each of these subproblems, since the dimension of each problem is relatively small, and BFGS gives a superlinear convergence rate while using only gradient information. When r in (4.9) is continuously differentiable, we can also use BFGS as our solver, see Algorithm 9. When r is non-smooth but admits an efficient prox operator, a first order method such as the proximal gradient method or its accelerations, such as FISTA [36], can be used instead, see Algorithm 10. We let ν denote the iteration counter.

---

[3] A function g is coercive if it grows in every direction, i.e. $\lim_{\alpha \uparrow \infty} g(\alpha x) = \infty$ for any $x \neq 0$.

Algorithm 9 VP using BFGS for outer problem (smooth r).

Input: $\boldsymbol{\alpha}^0$, $\mathbf{B}^0$, $\boldsymbol{w}^0$, $H_{\boldsymbol{\alpha}}^0 = I$, $\nu = 0$.

while not converged **do**
  for $j = 1, \ldots, n$ **do**
    $\boldsymbol{b}_j^{\nu+1} \leftarrow \arg\min_{\boldsymbol{b}} w_j^\nu \rho\left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha}^\nu)\boldsymbol{b}\right) + q(\boldsymbol{b})$
  end for
  $\boldsymbol{w}^{\nu+1} \leftarrow$ weights update
  $f_{\boldsymbol{\alpha}}^\nu \leftarrow f(\boldsymbol{\alpha}^\nu, \mathbf{B}^{\nu+1}, \boldsymbol{w}^{\nu+1})$
  $g_{\boldsymbol{\alpha}}^\nu \leftarrow \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^\nu, \mathbf{B}^{\nu+1}, \boldsymbol{w}^{\nu+1})$
  if $\nu \geq 1$ **then**
    $s^\nu \leftarrow f_{\boldsymbol{\alpha}}^\nu - f_{\boldsymbol{\alpha}}^{\nu-1}$
    $y^\nu \leftarrow g_{\boldsymbol{\alpha}}^\nu - g_{\boldsymbol{\alpha}}^{\nu-1}$
    $\beta^\nu \leftarrow (\langle s^\nu, y^\nu \rangle)^{-1}$
    $H_{\boldsymbol{\alpha}}^\nu \leftarrow \left[I - \beta^\nu (s^\nu)(y^\nu)^\top\right] H^{\nu-1} \left[I - \beta^\nu (y^\nu)(s^\nu)^\top\right] + \beta (s^\nu)(s^\nu)^\top$
  end if
  $\boldsymbol{\alpha}^{\nu+1} \leftarrow \text{LineSearch}(\boldsymbol{\alpha}^\nu - \eta_{\boldsymbol{\alpha}} H_{\boldsymbol{\alpha}}^\nu g_{\boldsymbol{\alpha}}^\nu)$
  $\nu \leftarrow \nu + 1$
end while

Output: $\boldsymbol{\alpha}^\nu$, $\mathbf{B}^\nu$.

Algorithm 10 VP using prox-gradient for outer problem (prox-friendly r).

Input: $\boldsymbol{\alpha}^0$, $\mathbf{B}^0$, $\boldsymbol{w}^0$, $\nu = 0$.

while not converged **do**
  for $j = 1, \ldots, n$ **do**
    $\boldsymbol{b}_j^{\nu+1} \leftarrow \arg\min_{\boldsymbol{b}} w_j^\nu \rho\left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha}^\nu)\boldsymbol{b}\right) + q(\boldsymbol{b})$
  end for
  $\boldsymbol{w}^{\nu+1} \leftarrow$ weights update
  $\boldsymbol{\alpha}^{\nu+1} \leftarrow \text{prox}_{\eta_{\boldsymbol{\alpha}} r}\left(\boldsymbol{\alpha}^\nu - \eta_{\boldsymbol{\alpha}} \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^\nu, \mathbf{B}^{\nu+1}, \boldsymbol{w}^{\nu+1})\right)$
  $\nu \leftarrow \nu + 1$
end while

Output: $\boldsymbol{\alpha}^\nu$, $\mathbf{B}^\nu$.

Updating $\boldsymbol{b}_j$ can be done efficiently by exploiting the optimized DMD problem structure. In particular, BFGS builds a Hessian approximation as it proceeds. All of the subproblems for $\boldsymbol{b}_j$ share the same $\boldsymbol{\Phi}(\boldsymbol{\alpha})$ and, since the columns of $\mathbf{B}$ contain spatial information, neighboring columns are likely to be similar to each other. After solving subproblem j, we use the resulting Hessian approximation to initialize the next subproblem $j+1$. Warm starts cut total BFGS iterations in half, see Figure 4.2.

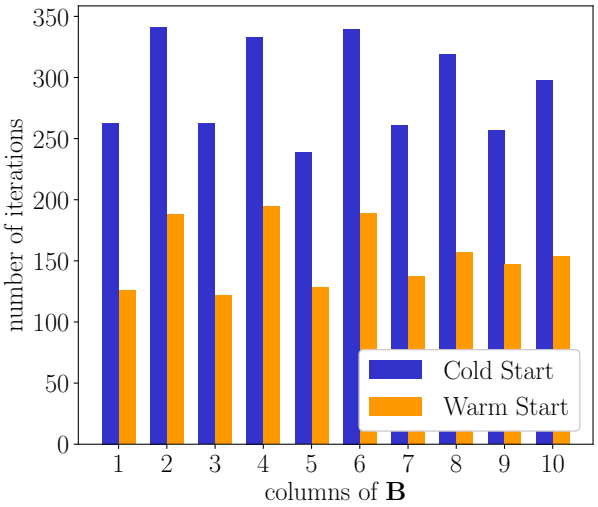


Figure 4.2: Average BFGS iterations for each subproblem across the columns.

There are different ways to update the weights $\boldsymbol{w}$, see line 4 in Algorithms 9 and 10. Define

$$\rho_j^\nu = \rho\left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha}^\nu)\boldsymbol{b}_j^{\nu+1}\right).$$

The objective with respect to $\boldsymbol{w}$ is given by

$$\min_{\boldsymbol{w}} \sum_{j=1}^{n} w_j \rho_j^\nu + s(\boldsymbol{w}),$$

where s encodes the weight constraints (4.11). The simplest update rule is to set $w_j = 1$ if ρ_j^ν is one of the h smallest, and 0 otherwise [364]; this corresponds to partial minimization in $\boldsymbol{w}$ at every step. A less aggressive strategy is to use proximal updates on $\boldsymbol{w}$,

$$\boldsymbol{w}^{\nu+1} = \text{prox}_{\eta_{\boldsymbol{w}} s}\left(\boldsymbol{w}^\nu - \eta_{\boldsymbol{w}} \nabla_{\boldsymbol{w}} f(\boldsymbol{\alpha}^\nu, \mathbf{B}^{\nu+1}, \boldsymbol{w}^\nu)\right)$$

where any step size $\eta_{\boldsymbol{w}} > 0$ can be used [12]. We use the former simple rule as the default in the algorithm. When $h = n$, trimming is turned off, and all weights are identically equal to 1.

4.3.3 A scalable stochastic method

In DMD applications, n represents the number of spatial variables, and is often much larger than either m or k. Therefore, step 2 of Algorithms 9 and 10 is a computational bottleneck. We use stochastic methods to scale the approach. The basic idea is to partially minimize

Algorithm 11 SVRG for DMD

Input: $\boldsymbol{\alpha}^0$, $\mathbf{B}^0$, $\boldsymbol{w}^0$

Initialize $\nu = 0$, $\zeta_j = \nabla f_j\left(\boldsymbol{\alpha}^0, \boldsymbol{w}^0\right)$ for $j = 1, 2, \ldots, n$, and $\zeta = \frac{1}{n}\sum_{j=1}^{n} \zeta_j$
while not converged **do**
  Uniformly sample $I^\nu \subset \{1, 2, \ldots, n\}$, such that $|I^\nu| = \tau$
  Sample $J^\nu \in \{0, 1\}$, such that $P(J = 1) \ll P(J = 0)$.
  for $j \in I_\nu$ **do**
    $\boldsymbol{b}_j^{\nu+1} \leftarrow \arg\min_{\boldsymbol{b}} w_j^\nu \rho(X_{\cdot,j} - \boldsymbol{\Phi}(\boldsymbol{\alpha}^\nu)\boldsymbol{b}) + q(\boldsymbol{b})$
    $\zeta_j^+ \leftarrow \nabla \tilde{g}_j\left(\boldsymbol{\alpha}^\nu, \boldsymbol{w}\right)$
  end for
  if $J = 1$ **then**
    $\boldsymbol{w}^{\nu+1} \leftarrow$ weights update
  else
    $\boldsymbol{w}^{\nu+1} \leftarrow \boldsymbol{w}^\nu$
  end if
  $\boldsymbol{\alpha}^{\nu+1} \leftarrow \mathrm{prox}_{\eta_{\boldsymbol{\alpha}} r}\left(\boldsymbol{\alpha}^\nu - \eta_{\boldsymbol{\alpha}}\left[\frac{1}{\tau}\sum_{j\in I^\nu}\left(\zeta_j^+ - \zeta_j\right) + \zeta\right]\right)$
  $\eta_{\boldsymbol{\alpha}} \leftarrow$ step size update
  $\zeta_j \leftarrow \zeta_j^+$ for $j \in I^\nu$
  $\zeta \leftarrow \frac{1}{n}\sum_{j=1}^{n} \zeta_j$
  $\nu \leftarrow \nu + 1$
end while

Output: $\boldsymbol{\alpha}^\nu$, $\mathbf{B}^\nu$

over a random sample of the columns of $\mathbf{B}$; the resulting (scaled) gradient is an unbiased estimate of $\nabla_{\boldsymbol{\alpha}} \tilde{f}$. More precisely, define

$$\begin{aligned}
\boldsymbol{b}_j(\boldsymbol{\alpha}, \boldsymbol{w}) &= \operatorname*{argmin}_{\boldsymbol{b}} w_j \rho\left(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{b}\right) + q(\boldsymbol{b}), \\
\tilde{g}_j(\boldsymbol{\alpha}, \boldsymbol{w}) &= w_j \rho(X_{\cdot j} - \boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{b}_j(\boldsymbol{\alpha}, \boldsymbol{w})) + q(\boldsymbol{b}_j(\boldsymbol{\alpha}, \boldsymbol{w})).
\end{aligned}$$

Then we have

$$\tilde{f}(\boldsymbol{\alpha}, \boldsymbol{w}) = \sum_{j=1}^{n} \tilde{g}_j(\boldsymbol{\alpha}, \boldsymbol{w}) + r(\boldsymbol{\alpha}) + s(\boldsymbol{w}).$$

This is a classical setting for stochastic methods. In each iteration, we can use a subset of $\tilde{g}_j$ to calculate the approximate gradient for the smooth part of $\tilde{f}$ in order to reduce the computational burden. Here we use SVRG [198] as our stochastic solver for the outer problem; the full details are given in Algorithm 11. Note that this stochastic approach is an alternative to using a cost reduction based on projecting onto SVD modes [20] or using an optimized but fixed subsampling of the columns [175]. With the method of Algorithm 11, none of the data is discarded or filtered by the cost reduction procedure.

In Figure 4.3, we solve a problem with dimension $m = 512$ and $n = 1000$. A diminishing step size scheme is used, taking

$$\eta_{\boldsymbol{\alpha}}^{\nu} = \frac{\eta_{\boldsymbol{\alpha}}^{0}}{\text{floor}(\nu/K) + 1},$$

with $\eta_{\boldsymbol{\alpha}}^{0} = 10^{-7}$ and $K = 500$ for the result in Figure 4.3. Comparing the algorithms according to total $\boldsymbol{b}_j$ subproblems, we see that SVRG decreases faster and is less noisy than the Stochastic Proximal Gradient (SPG) method[4]. Proximal Gradient (PG) decreases quickly in the beginning, but is soon overtaken by stochastic methods. SVRG gives a significant improvement over SPG.

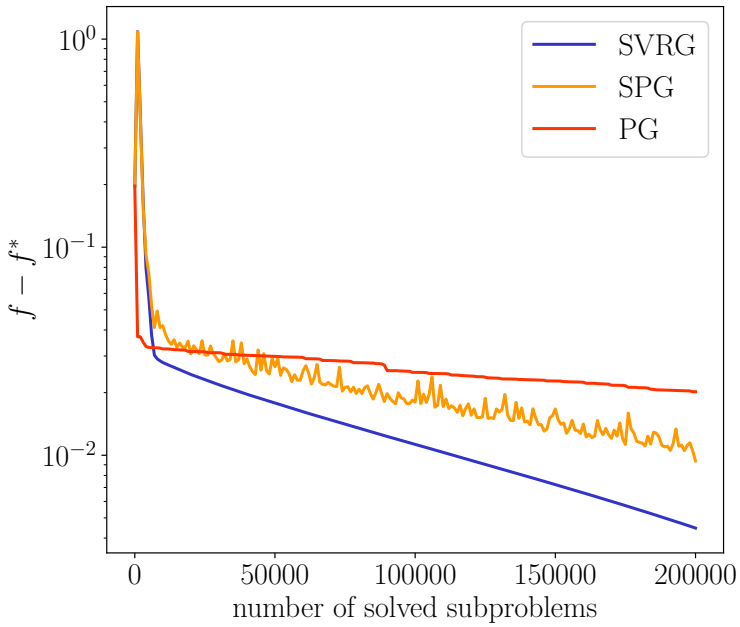


Figure 4.3: Compare performance of SVRG, Stochastic Proximal Gradient (SPG) method and Proximal Gradient (PG) method over the same data set.

The trimming weights $\boldsymbol{w}$ rely on global information; that is, the best h residuals are easily selected after all of the residuals have been calculated. This is why the weights update (lines 8-11 of Algorithm 11) is done rarely. For detailed analysis of stochastic algorithms with trimming, see [12].

[4]It is important to note that SPG has no convergence theory, while SVRG is guaranteed to converge. In practice SPG works well so we include it in the comparison.

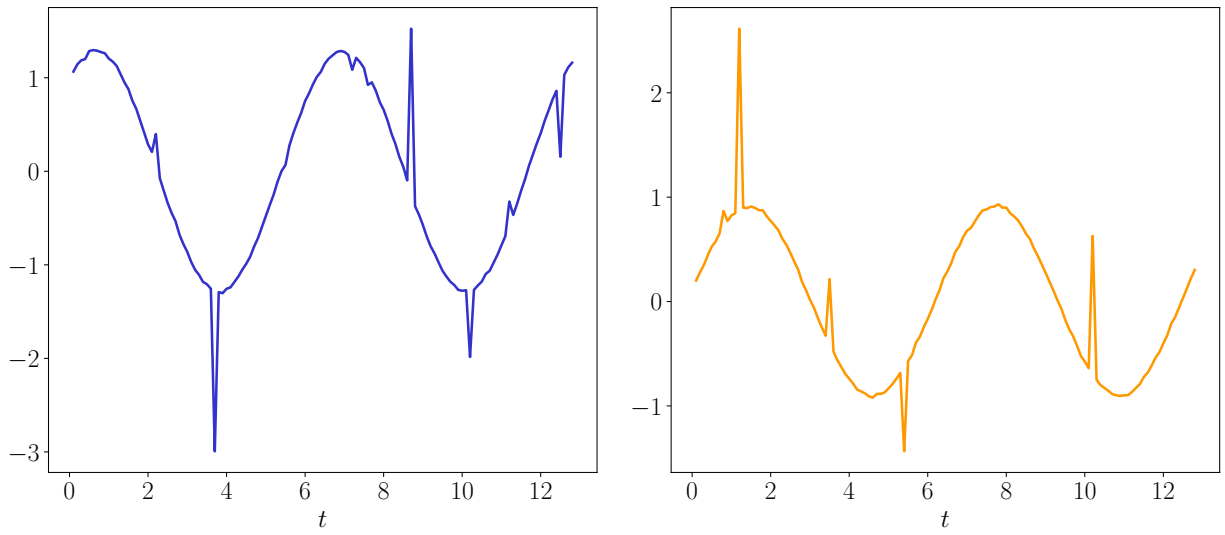


Figure 4.4: Sample time series of $x_1(t)$ and $x_2(t)$ for the simple periodic example, with background noise of size $\sigma = 10^{-2}$ and spikes of size $\mu = 1$ added at $p = 5\%$ of the snapshots for each channel.

4.4 Synthetic examples

In this section, we examine the performance of the robust DMD on a pair of synthetic test cases with known solution. These examples are drawn from the additive noise study of [109].

4.4.1 A simple periodic example

Let $\boldsymbol{x}(t)$ be the solution of a two dimensional linear system with the following dynamics

$$\dot{\boldsymbol{x}} = \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix} \boldsymbol{x} \, . \tag{4.17}$$

We use the initial condition $\boldsymbol{x}(0) = (1, 0.1)^\top$ and take snapshots

$$\boldsymbol{x}_j = \boldsymbol{x}(j\Delta t) + \sigma \mathbf{g}_j + \mu \mathbf{s}_j \, ,$$

where $\Delta t = 0.1$, σ and μ are prescribed noise levels, $\mathbf{g}_j$ is a vector whose entries are drawn from a standard normal distribution, and $\mathbf{s}_j$ is a vector whose entries are the product of a Bernoulli trial with small expectation p and a standard normal (corresponding to sparse noise). The snapshots are therefore corrupted with a base level of noise σ and sparse "spikes" of size μ with firing rate p. A sample time series for this example can be found in Figure 4.4.

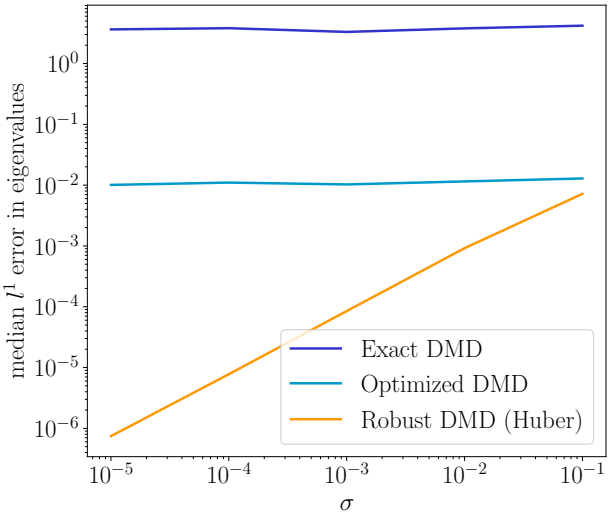


Figure 4.5: Median error in the computed eigenvalues over 200 runs. The background noise σ varies while the size of the spikes is fixed at $\mu = 1$ and the firing rate is fixed at $p = 5\%$.

The $k = 2$ eigenvalues of the system matrix in (4.17) are $\pm\boldsymbol{i}$, corresponding to sinusoidal dynamics in time. In Figure 4.5, we plot the median (over 200 random trials) of the l^1-norm error in the approximations of these eigenvalues using three different methods: the exact DMD of [337]; the optimized DMD as defined in (4.4); and the robust DMD as defined in (4.9), with ρ the Huber norm and $h = n = 2$ (no trimming). Each trial consists of the first 128 snapshots with additive noise. The level of the background noise, σ, varies over the experiments and the size and firing rate of the spikes are fixed at $\mu = 1$ and $p = 5\%$, respectively. We set the Huber parameter using knowledge of the problem set-up, i.e. $\kappa = 5\sigma$, but in a real-data setting this parameter would have to be estimated or chosen adaptively. While the optimized DMD improves over the exact DMD, the error does not decrease as the level of the background noise decreases. We therefore see the effect of the sparse outliers using the optimized DMD. For the robust formulation, on the other hand, the accuracy of the eigenvalues is determined by the level of the background noise, so that the outliers are not biasing the computed eigenvalues.

4.4.2 An example with hidden dynamics

In the case that a signal contains some rapidly decaying components it can be more difficult to identify the dynamics, particularly in the presence of sensor noise [109]. We consider a signal composed of two sinusoidal forms which are translating, with one growing and one decaying, i.e.

$$x(y,t) = \sin(k_1 y - \omega_1 t)e^{\gamma_1 t} + \sin(k_2 y - \omega_2 t)e^{\gamma_2 t}\,, \tag{4.18}$$

where $k_1 = 1$, $\omega_1 = 1$, $\gamma_1 = 1$, $k_2 = 0.4$, $\omega_2 = 3.7$, and $\gamma_2 = -0.2$ (following settings used by [109]). This signal has $k = 4$ continuous time eigenvalues given by $\gamma_1 \pm i\omega_1$ and $\gamma_2 \pm i\omega_2$. We set the domain of y to be $[0, 15]$ and use 300 equispaced points, y_j, to discretize. For the time domain, we set $\Delta t = \pi/(2^8 - 2)$ so that the number of snapshots we use, $m = 2^7$, covers $[0, \pi/2]$. We denote the vector of discrete values $x(y_j, t)$ by $\boldsymbol{x}(t)$. See Figure 4.6a for a surface plot of this data.

We consider three different types of perturbations of the data. The first perturbation adds background noise and spikes, as in the previous example, i.e. the snapshots are given by

$$\boldsymbol{x}_j^{(1)} = \boldsymbol{x}(j\Delta t) + \sigma \mathbf{g}_j + \mu \mathbf{s}_j\,,$$

where σ and μ are prescribed noise levels, $\mathbf{g}_j$ is a vector whose entries are drawn from a standard normal distribution, and $\mathbf{s}_j$ is a vector whose entries are the product of a Bernoulli trial with small expectation p and a standard normal. See Figure 4.6b for a sample plot of this "sparse noise" pattern. The second perturbation we consider adds background noise and spikes which are confined to specific entries of $\boldsymbol{x}_j$, i.e.

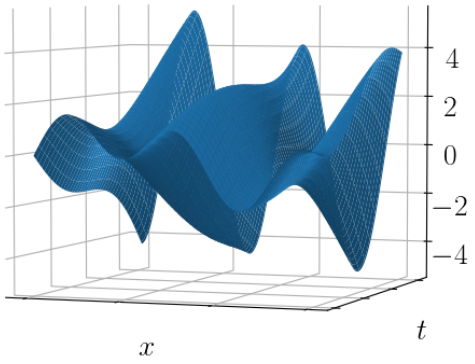
$$\boldsymbol{x}_j^{(2)} = \boldsymbol{x}(j\Delta t) + \sigma \mathbf{g}_j + \mu \tilde{\mathbf{s}}_j\,,$$

where $\mathbf{g}_j$, σ, and μ are as above and the $\tilde{\mathbf{s}}_j$ are sparse vectors which have the same sparsity pattern for all j and nonzero entries drawn from a standard normal distribution (this corresponds to having a few broken sensors recording the data). We plot a sample of this "broken sensor" noise pattern in Figure 4.6c. The third perturbation we consider adds background noise and a localized bump to the data, i.e.

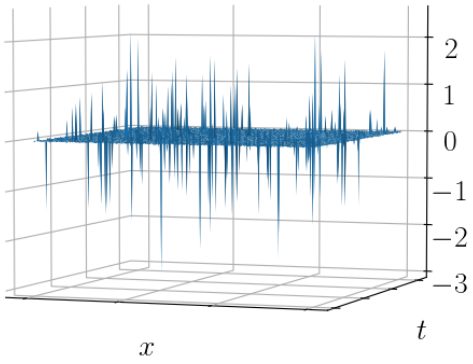
$$\left[\boldsymbol{x}_j^{(3)}\right]_i = x(y_i, j\Delta t) + \sigma\mathcal{N}(0,1) + A\exp\left(-\left(\frac{y_b - y_i}{w\Delta y}\right)^2 - \left(\frac{t_b - j\Delta t}{w\Delta t}\right)^2\right)\,,$$

where σ is as above, $\mathcal{N}(0,1)$ denotes a number drawn from the standard normal distribution, A determines the maximum height of the bump, w determines the "width" of the bump, and y_b and t_b determine the center of the bump in space and time (this corresponds to having some non-exponential dynamics in the data). In Figure 4.6d, we plot a sample of this "bump" noise pattern.

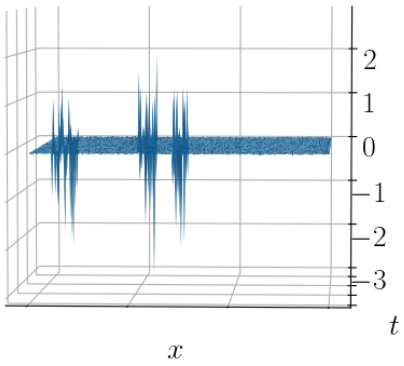
In Figure 4.7, we plot the median (over 200 random trials) of the l^1-norm error in the approximations of the eigenvalues using four different methods: the exact DMD of [337]; the optimized DMD as defined in (4.4); the robust DMD as defined in (4.9), with ρ the Huber norm and $h = n = 300$ (no trimming); and the robust DMD with ρ the standard Frobenius norm and $h = 0.8n = 240$ (trimming). Each trial consists of the first 128 snapshots with additive noise. The level of the background noise, σ, varies over the experiments. For the "sparse noise" and "broken sensor" snapshots, the size of the spikes is fixed at $\mu = 1$ and



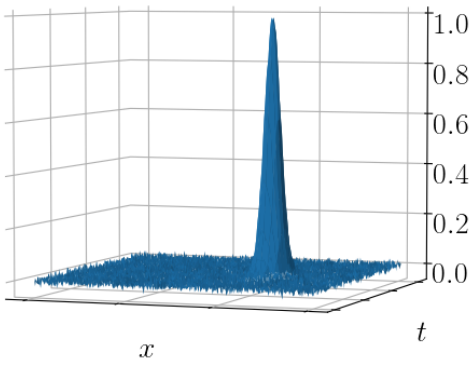
(a) clean data



(b) “sparse noise”

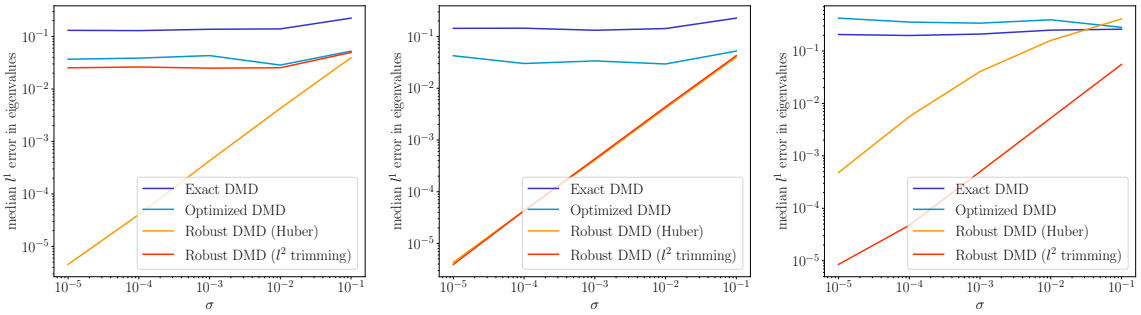


(c) “broken sensor”



(d) “bump”

Figure 4.6: A surface plot of the data for the hidden dynamics example and surface plots of a sample of each type of noise we consider.



(a) "sparse noise" (b) "broken sensor" (c) "bump"

Figure 4.7: Median error in the computed eigenvalues over 200 runs. The background noise σ varies while the size of the spikes is fixed at $\mu = 1$ and the firing rate is fixed at $p = 5\%$ for the "sparse noise" and "broken sensor" examples and the height is fixed at $A = 1$ and the width at $w = 10$ for the "bump" example.

the density is fixed at $p = 5\%$, i.e. 5% of the entries are corrupted for the "sparse noise" example and 5% of the sensors are corrupted for the "broken sensor" example. For the "bump" snapshots, the height of the bump is fixed at $A = 1$ and the width at $w = 10$. We set the Huber parameter using knowledge of the problem set-up, i.e. $\kappa = 5\sigma$, but in a real-data setting this parameter would have to be estimated or chosen adaptively.

With sparse noise, as in Figure 4.7a, the results for the exact DMD, optimized DMD, and Huber norm-based robust DMD are consistent with the simple periodic example. The Huber norm formulation is the only one which is able to take advantage of the lower levels of background noise. The trimming formulation provides very little advantage for this example, as any sensor can be affected by the outliers. In contrast, we see that the trimming formulation is able to perform as well as the Huber formulation for the broken sensor example (see Figure 4.7b), as the algorithm is able to adaptively remove the broken sensors from the data. In Figure 4.7c, we plot the results for the bump data, which display some interesting behavior. The optimized DMD actually performs worse than the exact DMD, which is attributable to over-fitting. For all but the highest background noise level, the Huber and trimming formulations show a significant advantage over the optimized DMD and exact DMD, with the trimming formulation performing the best. The trimming formulation therefore presents an attractive solution for data with unknown, localized deviations from the exponential basis of the DMD, especially given that the inner problem for trimming with the Frobenius penalty can be solved rapidly. Of course, trimming can be combined with a Huber (or other robust) penalty for increased robustness to outliers.

4.5 Conclusion and future directions

We have presented an optimization framework and a suite of numerical algorithms for computing the dynamic mode decomposition with robust penalties and parameter constraints. This framework allows for improved performance of the DMD in a number of settings, as borne out by synthetic and real data experiments. In the presence of sparse noise or non-exponential structure, the use of robust penalties significantly decreases the bias in the computed eigenvalues. When using the DMD to perform future state prediction, adding the constraint that the eigenvalues lie in the left half-plane increases the stability of the extrapolation. The algorithms presented are capable of solving small to medium-sized problems in seconds on a laptop and scale well to higher-dimensional problems due to their intrinsic parallelism and the efficiency of the SVRG approach. In contrast with previous approaches, the SVRG increases efficiency without throwing out data or incidentally filtering it. We believe that the framework and algorithms presented here will enable practitioners of the DMD to tackle larger, noisier, and more complex data sets than previously possible. The authors commit to releasing the software used for these calculations as an open-source package in the Julia language [44].

The present work can be extended in a number of ways. Because the inner solve completely decouples over the columns of $\mathbf{X}$ and $\mathbf{B}$, the algorithms presented above immediately generalize to data-sets with missing entries and even data which are collected asynchronously across sensors. While the global nature of an optimized DMD fit has advantages in terms of the quality of the recovered eigenvalues, it implicitly rules out process noise. However, including process noise or a known forcing term would be useful in many applications. Incorporating such terms into this optimization framework is ongoing work and results will be reported at a later date. We also note that much of the above applies to dimensionality reduction using any parameterized family of time dynamics, not just exponentials. For such an application, many of the algorithms above could be easily adapted, so long as gradient formulas are available.

Chapter 5

ROBUST SPARSE PRINCIPLE COMPONENT ANALYSIS

Sparse principal component analysis (SPCA) has emerged as a powerful technique for modern data analysis, providing improved interpretation of low-rank structures by identifying localized spatial structures in the data and disambiguating between distinct time scales. We demonstrate a robust and scalable SPCA algorithm by formulating it as a value-function optimization problem. This viewpoint leads to a flexible and computationally efficient algorithm. It can further leverage randomized methods from linear algebra to extend the approach to the large-scale (big data) setting. Our proposed innovation also allows for a robust SPCA formulation which can also obtain meaningful sparse principal components in spite of grossly corrupted input data. The proposed algorithms are demonstrated using both synthetic and real world data, showing exceptional computational efficiency and diagnostic performance.

5.1 Introduction

A wide range of phenomena in the physical, engineering, biological, and social sciences feature rich dynamics that give rise to multiscale structures in both space and time, including fluid dynamics, atmospheric-ocean interactions, climate modeling, epidemiology, and neuroscience. Remarkably, the underlying dynamics of such systems are typically inherently low-rank in nature, generating data sets where dimensionality reduction techniques, such as principal component analysis (PCA), can be used as a critically enabling diagnostic tool for interpretable characterizations of the dynamics. PCA decompositions express time-varying patterns as a linear combination of the dominant correlated spatial activity of the state of a system as it evolves in time. Although commonly used, the PCA approach generates global modes that often mix or blend various spatio-temporal scales, and cannot identify underlying governing dynamics that act at separate scales. Moreover, classic PCA also tends to overfit data where the number of observations is smaller than the number of variables [72].

Constrained or regularized matrix decompositions provide a more flexible approach for modeling dynamic patterns. Specifically, prior information can be introduced through sparsity promoting regularizers to obtain a more parsimonious approximation of the data which typically provides improved interpretability. Among others, *sparse principal component analysis* (SPCA) has emerged as a popular and powerful technique for modern data analysis. SPCA promotes sparsity in the modes, i.e., the sparse modes have only a few *active* coefficients, while the majority of coefficients are constrained to be zero. The resulting sparse modes are often highly localized and more interpretable than the global PCA modes obtained

from traditional PCA. As a consequence, sparse regularization of PCA allows for a decomposition strategy that can specifically identify localized spatial structures in the data and disambiguate between distinct time scales, both of which are ubiquitous in measurement data of complex systems. As an example, one only needs to consider the physical phenomenon of El Nino which is a mode characterized by a localized warm temperature profile which traverses the southern Pacific ocean. This is a highly localized mode that, as will be shown, is well characterized by SPCA, while standard PCA gives a global mode with nonzero values across the entire globe.

While the idea of sparsifying the weight vectors is not new, simple ad-hoc techniques such as naive thresholding can lead to misleading results. A formal approach to SPCA, using ℓ_1 regularization, was first proposed by Jolliffe et al. [200]. This pioneering work lead to a variety of sparsity promoting algorithms [387, 105, 128, 313, 311, 357, 201]. The success of sparse PCA in obtaining interpretable modes motivates the general approach developed in this paper. Specifically, our method offers three immediate improvements over previously proposed SPCA algorithms: (1) a faster and scalable algorithm, (2) robustness to outliers, and (3) straightforward extension to nonconvex regularizers, including the ℓ_0 norm. Scalability is essential for many applications — for example, dynamical systems generate very large-scale datasets, such as the sea surface temperature data analyzed in this paper. Robust formulations allow SPCA to be deployed in a broader setting, where data contamination could otherwise hide sparse modes. Nonconvex regularizers are not currently available in SPCA software — we show that the modes we get with these approaches are better in synthetic examples, and more interpretable for real data.

Contributions of this work In this work, we develop a scalable and robust approach for SPCA. A key feature of the approach is the use of *variable projection* to partially minimize over the orthogonally constrained variables. This idea was used in the original alternating approach of [387], and we innovate on this idea by recasting the problem as a value-function optimization. This viewpoint allows for significantly faster algorithms, scalability, and broader applicability. We also allow nonconvex regularization on the loadings, which further improves interpretability and sparsity. Not only does the method scale well, but it is further accelerated using randomized methods for linear algebra [138]. Further, the proposed approach extends to robust SPCA formulations, which can obtain meaningful principal components even with grossly corrupted input data. The outliers are modeled as perturbations to the data, as in the robust PCA model [79, 53, 9]. These innovations provide a flexible and highly-efficient algorithm for modern data analysis and diagnostics that will enable a wide range of critical applications at a scale not previously possible with other leading algorithms.

Organization The manuscript is organized as follows: Section 5.2 reviews PCA and the variable projection framework. Section 5.3 provides a detailed problem formulation and discusses the variable projection viewpoint which is advocated in this paper. Further, different loss functions and regularizes are discussed. We present the details of the proposed

algorithms in Section 5.4. First, the standard case, using the least squares loss function, is discussed. Then, a randomized accelerated and a robust algorithm is presented. The method is applied to several examples in Section 5.6, where SPCA correctly identifies dynamics occurring at different timescales in multiscale data. We draw conclusions about the method and discuss its outlook in Section 5.7.

Notation Scalars are denoted by lower case letters x, and vectors in $\mathbb{R}^n$ are denoted as bold lower case letters $\mathbf{x} = [x_1, x_2, ..., x_n]^\top$. Matrices are denoted by bold capital letters $\mathbf{M}$. The transpose of a real matrix is denoted as $\mathbf{M}^\top$. The spectral or operator norm of a matrix is denoted as $\|\cdot\|$ and the Frobenius norm is denoted as $\|\cdot\|_F$.

5.2 Background

5.2.1 Principal Component Analysis

Principal component analysis (PCA) is an ubiquitous dimension reduction technique, tracing back to Pearson [276] and Hotelling [192]. The aim of PCA is to find a set of new uncorrelated variables, called principal components (PCs), such that the first PC accounts for the greatest amount of variance in the data, the second PC for the second greatest variance, and so on. More concretely, let $\mathbf{X}$ be a real data matrix of dimension $n \times p$, with column-wise zero empirical mean. The n rows represent observations and the p columns correspond to measurements of variables. The principal components $\mathbf{z}_i \in \mathbb{R}^n$ are formed as a linear weighted combination of the variables

$$\mathbf{z}_i = \mathbf{X}\mathbf{a}_i, \tag{5.1}$$

where $\mathbf{a}_i \in \mathbb{R}^p$ is a vector of weights. This can be expressed more concisely as

$$\mathbf{Z} = \mathbf{X}\mathbf{A}, \tag{5.2}$$

with $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_p] \in \mathbb{R}^{n\times p}$ and $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_p] \in \mathbb{R}^{p\times p}$. The orthonormal matrix $\mathbf{W}$ rotates the data into a new space, where the principal components sequentially capture the maximum variability in the input data. The columns of $\mathbf{A}$ are also often denoted as modes, basis functions, principal direction or loadings.

Mathematically, a variance maximization problem can be formulated to find the weight vectors $\mathbf{a}_i$. Alternatively, the problem can be formulated as a least-squares problem, i.e., minimizing the sum of squared residual errors between the input and the projected data

$$\begin{aligned} \underset{\mathbf{A}}{\text{minimize}} \quad & f(\mathbf{A}) = \|\mathbf{X} - \mathbf{X}\mathbf{A}\mathbf{A}^\top\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \mathbf{A}^\top\mathbf{A} = \mathbf{I}, \end{aligned} \tag{5.3}$$

where PCA imposes orthogonality constraints on the weight matrix $\mathbf{A}$. Given the singular value decomposition (SVD) of the centered (standardized) input matrix $\mathbf{X}$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

the minimizer of (5.3) is given by the right singular vectors $\mathbf{V}$, i.e., we can set $\mathbf{A} = \mathbf{V}$. Further, the principal components are the scaled left singular vectors $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}$, where the entries of the diagonal matrix $\mathbf{\Sigma}$ are the singular values. In most applications, we are only interested in the first k dominant PCs which account for most of the variability in the input data. Thus, PCA allows one to reduce the dimensionality from p to k by simply truncating the SVD. The dominant k PCs can be used to visualize the data in low-dimensional space, and as features for clustering, classification and regression.

We refer the reader to [199] for an extensive treatment of PCA and its mechanics. Many extensions such as Kernel PCA have been proposed to extend and overcome some of the shortcomings of PCA, see [103] for a brief overview.

5.2.2 Variable Projection

Consider any objective of the form

$$\min_{\mathbf{A},\mathbf{B}} g(\mathbf{A}, \mathbf{B}). \tag{5.4}$$

A classic example is the nonlinear least squares problem

$$\min_{\mathbf{A},\mathbf{B}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{\Phi}(\mathbf{B})\mathbf{A}\|^2. \tag{5.5}$$

The term 'variable projection' [165] originally arose from the fact that the least squares projection of $\mathbf{X}$ onto the range of $\mathbf{\Phi}(\mathbf{B})$ has a closed form solution, which is used explicitly in iterative methods to optimize for $\mathbf{B}$. More generally, the word 'projection' is now associated with epigraphical projection [295], or partial minimization. We can rewrite (5.4) as a value function optimization problem:

$$\min_{\mathbf{B}} \left\{ v(\mathbf{B}) := \min_{\mathbf{A}} g(\mathbf{A}, \mathbf{B}) \right\}. \tag{5.6}$$

In many cases, the function $v(\mathbf{B})$ has an explicit form. In the classic problem (5.5), we have

$$v(\mathbf{B}) = \frac{1}{2}\|\mathbf{X}(\mathbf{I} - \mathcal{P}_{\mathcal{R}(\mathbf{\Phi}(\mathbf{B}))})\|^2 = \mathrm{dist}^2(\mathbf{X}|\mathcal{R}(\mathbf{\Phi}(\mathbf{B}))),$$

where $\mathcal{P}_{\mathcal{R}(\mathbf{\Phi}(\mathbf{B}))}$ is a projector on the range of $\mathbf{\Phi}(\mathbf{B})$. Explicit expressions are not necessary as long as we have an efficient routine to compute

$$\mathbf{A}(\mathbf{B}) = \arg\min_{\mathbf{A}} g(\mathbf{A}, \mathbf{B}).$$

For many problems, we can find first and second derivatives of $v(\mathbf{B})$. For example, when g is smooth and $\mathbf{A}(\mathbf{B})$ is unique, we have

$$\nabla v(\mathbf{B}) = \partial_{\mathbf{B}} g(\cdot, \cdot)|_{(\mathbf{A}(\mathbf{B}),\mathbf{B})}.$$

Formulas for second derivatives are collected in [15]. Variable projection was recently used to solve a range of large-scale structured problems in PDE-constrained optimization, nuisance parameter estimation, exponential fitting, and optimized dynamic mode decomposition [8, 15, 191, 21].

5.3 Problem Formulation for Sparse Principal Component Analysis (SPCA)

Sparse PCA aims to find a set of sparse weight vectors, i.e., weight vectors with only a few 'active' (nonzero) values. In this manuscript, we build up on the seminal work by Zou, Hastie and Tibshirani [387], who treat SPCA as an regularized regression-type problem. More concretely, their formulation directly incorporates sparsity inducing regularizers into the optimization problem:

$$\begin{aligned} \underset{\mathbf{A},\mathbf{B}}{\text{minimize}} \quad & f(\mathbf{A},\mathbf{B}) = \tfrac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_{\mathrm{F}}^2 + \psi(\mathbf{B}) \\ \text{subject to} \quad & \mathbf{A}^\top\mathbf{A} = \mathbf{I}, \end{aligned} \tag{5.7}$$

where $\mathbf{B}$ is a sparse weight matrix and $\mathbf{A}$ is an orthonormal matrix. ψ denotes a sparsity inducing regularizer such as the LASSO (ℓ_1 norm) or the elastic net (a combination of the ℓ_1 and squared ℓ_2 norm). The optimization problem is minimized using an alternating algorithm:

- **Update A.** With $\mathbf{B}$ fixed, we find an orthonormal matrix $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$ which minimizes

  $$\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_F^2.$$

  This is the orthogonal Procrustes problem [171] (see Appendix 5.8.1), which has a closed form solution $\mathbf{A}^* = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{X}^\top\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.

- **Update B.** With $\mathbf{A}$ fixed, we solve the optimization problem

  $$\min_{\mathbf{B}} \ \tfrac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_{\mathrm{F}}^2 + \psi(\mathbf{B}).$$

  The problem splits across the k columns of $\mathbf{B}$, yielding a regularized regression problem in each case:

  $$\mathbf{b}_j^* = \underset{\mathbf{b}_j}{\operatorname{argmin}} \tfrac{1}{2}\|\mathbf{X}\mathbf{A}(:,j) - \mathbf{X}\mathbf{b}_j\|^2 + \psi(\mathbf{b}_j).$$

The principal components are then formed as a sparsely weighted linear combination of the observed variables $\mathbf{Z} = \mathbf{X}\mathbf{B}$. The data can be approximately rotated back as $\widetilde{\mathbf{X}} = \mathbf{Z}\mathbf{A}^\top$.

Coordinate descent or least angle regression (LARS) are used to solve each of the k subproblems [129]. The $\mathbf{B}$ update relies on solving a strongly convex problem, and in particular the update is unique, and the algorithm as described converges to as stationary point by the analysis of [336]. Replacing ψ with a nonconvex regularizer, such as $\psi(\mathbf{B}) = \alpha\|\mathbf{B}\|_0 + \beta\|\mathbf{B}\|^2$, makes it difficult to guarantee anything about the $\mathbf{B}$ update. However, as we show, using the value function (5.6) from the variable projection viewpoint yields an efficient implementation and a straightforward convergence analysis.

5.3.1 Variable Projection Viewpoint

The $\mathbf{A}$ update in the method of [387] is in closed form, while the $\mathbf{B}$ update requires an iterative method. To exploit the efficiency of the $\mathbf{A}$ update, we think of projecting out $\mathbf{A}$ and introduce the sparse PCA value function

$$v(\mathbf{B}) := \min_{\mathbf{A}} \quad \tfrac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_{\mathrm{F}}^2 \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I},$$

viewing the original SPCA problem (5.7) as

$$\min_{\mathbf{B}} v(\mathbf{B}) + \psi(\mathbf{B}). \tag{5.8}$$

We show that $v(\mathbf{B})$ is differentiable with a Lipschitz continuous gradient, and derive its explicit form. This viewpoint lets us use any proximal algorithm we want to minimize (5.8), including proximal gradient (see e.g. [275]) and FISTA [36], with the caveat that an $\mathbf{A}$ update is computed every time $v(\mathbf{B})$ is evaluated. For the original SPCA problem, this approach rebalances the work between the $\mathbf{A}$ and $\mathbf{B}$ updates, using a single operator to update $\mathbf{B}$ instead of an iterative routine. When combined with randomized techniques for computing the $\mathbf{A}$ update, we get an order of magnitude acceleration compared to current SPCA software.

The variable projection viewpoint (5.8) also allows a robust SPCA approach with the Huber loss function. Simply replacing the quadratic penalty in (5.7) with a different loss would destroy the efficient structure of the $\mathbf{A}$ update, requiring an iterative routine to solve for it. Instead, we use a special characterization of the Huber function to obtain a formulation with three rather than two variables, preserving the efficiency of each update. We extend our analysis to this case, so the robust formulation can also be used with any prox-friendly ψ regularizer, including the nonconvex example discussed above.

In summary, the value function viewpoint also makes it easy to extend to a broader problem setting, and we consider the following objective:

$$\min_{\mathbf{A},\mathbf{B}} \quad f(\mathbf{A},\mathbf{B}) := \rho(\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top) + \psi(\mathbf{B}) \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}, \tag{5.9}$$

where ρ is a separable loss, while ψ is a separable regularizer for $\mathbf{B}$.

5.3.2 Regularizers for Sparsity

The SPCA framework incorporates a range of sparsity-inducing regularizers ψ. Sparsity is achieved by introducing additional information into the model to find the most meaningful 'active' (non-zero) entries in $\mathbf{B}$, while most of the loadings are constrained to be zero. Sparse approaches work well when many variables are redundant, i.e., not required to capture the underlying coherent model structure. Regularization also prevents overfitting, and provides a path to solve ill-posed problems, frequently encountered in the analysis of high-dimensional datasets.

5.3.3 *Unstructured Sparsity*

The ℓ_0 'norm', denoted $\ell_0(\boldsymbol{x})$ or $\|\boldsymbol{x}\|_0$, counts the number of non-zero elements in a vector $\mathbf{x}$. When used as a regularizer ψ, it encourages models with small cardinality, i.e., a small number of active loadings. Although ℓ_0 is non-smooth and non-convex, its proximal operator is simply hard thresholding (see Table 5.1).

In many applications, the ℓ_1 norm is used to approximate ℓ_0. In the context of least squares problems, using ℓ_1 is known as LASSO (least absolute shrinkage and selection operator). The proximal operator of the scaled ℓ_1 norm $\gamma\|\boldsymbol{x}\|_1$ is the *soft-thresholding* operator, see Table 5.1.

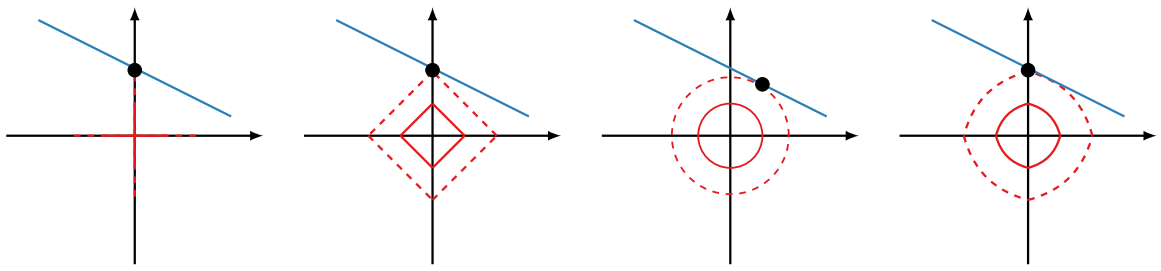
One drawback of the ℓ_1 norm is that it tends to activate only one coefficient from any set of highly correlated variables. The elastic net, introduced by Zou and Hastie [386], overcomes this drawback, using a linear combination of the ℓ_1 and quadratic penalty:

$$\psi_{12}(\boldsymbol{x}) = \alpha\|\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_2^2.$$

The elastic net has an implicit grouping effect that is particularly useful for the analysis of high-dimensional multiscale physical systems, where we want to find all the associated variables which correspond to an underlying mode, rather than selecting only one variable from each underlying mode. The proximal operator of ψ_{12} combines scaling and soft thresholding, see Table 5.1. Following the same idea, we can also combine ℓ_0 and the quadratic penalty:

$$\psi_{02}(\boldsymbol{x}) = \alpha\|\boldsymbol{x}\|_0 + \beta\|\boldsymbol{x}\|_2^2.$$

The ψ_{02} regularizer detects correlated sets of very sparse predictors, and its proximal operator of ψ_{02} combines scaling and hard thresholding, see Table 5.1. Figure 5.1 illustrates these regularizers. Many other examples of proximal operators are collected in [96].



(a) ℓ_0 norm. (b) ℓ_1 norm. (c) ℓ_2 norm. (d) Elastic net.

Figure 5.1: Illustration of some norms which are used as regularizers. ℓ_0, ℓ_1 and elastic net are sparsity-inducing.

	Regularizer ψ	$\text{prox}_{\gamma\psi}(\boldsymbol{x})_i$
ψ_0	$\|\boldsymbol{x}\|_0$	Hard Thresholding : $\begin{cases} x_i, & x_i^2 > 2\gamma \\ 0, & \text{otherwise} \end{cases}$
ψ_1	$\|\boldsymbol{x}\|_1$	Soft Thresholding : $\begin{cases} x_i - \gamma, & x_i > \gamma \\ x_i + \gamma, & x_i < -\gamma \\ 0, & \text{otherwise} \end{cases}$
ψ_{02}	$\alpha\|\boldsymbol{x}\|_0 + \beta\|\boldsymbol{x}\|_2^2$	Scaled Hard Thresholding : $\begin{cases} x_i/(1+2\gamma\beta), & x_i^2 > 2\gamma\alpha(1+2\gamma\beta) \\ 0, & \text{otherwise} \end{cases}$
ψ_{12}	$\alpha\|\boldsymbol{x}\|_1 + \beta\|\boldsymbol{x}\|_2^2$	Scaled Soft Thresholding : $\begin{cases} (x_i - \gamma\alpha)/(1+2\gamma\beta), & x > \gamma\alpha \\ (x_i + \gamma\alpha)/(1+2\gamma\beta), & x < -\gamma\alpha \\ 0, & \text{otherwise} \end{cases}$

Table 5.1: Regularizers ψ and their proximal operators.

5.3.4 Structured Sparsity

A large number of separable structured regularizers ψ can be used in the proposed SPCA framework. Separability ensures that the prox-operator can be computed either in closed form or using a routine for both convex and nonconvex regularizers. Here we highlight two examples.

In some applications, selection occurs between groups of variables known *a priori*. The group lasso regularizer [371] enforces that all the variables corresponding to these predefined group are either activated or set to 0. Its prox operator can be written as

$$\text{prox}_{\gamma\|\cdot\|_2}(\boldsymbol{x}) = \begin{cases} (1 - \gamma/\|\boldsymbol{x}\|_2)\boldsymbol{x}, & \|\boldsymbol{x}\|_2 > \gamma \\ \mathbf{0}, & \|\boldsymbol{x}\|_2 \leq \gamma \end{cases}.$$

An extension is the sparse group lasso [314], which adds an additional ℓ_1 penalty for each group. Another useful regularizer is the fused lasso [330], which gives a way to incorporate information about spatial or temporal structure in the data.

5.4 Fast Algorithms for Sparse PCA

5.4.1 Sparse PCA via Variable Projection

As a standard problem, we discuss the variable projection algorithm for (5.9) using the least squares loss function. We partially minimize in $\mathbf{A}$ to obtain the *value* function

$$v(\mathbf{B}) := \min_{\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_F^2 \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \tag{5.10}$$

Evaluating this value function given $\mathbf{B}$ reduces to solving the orthogonal Procrustes problem [171], with closed form solution

$$\mathbf{A}(\mathbf{B}) = \mathbf{U}\mathbf{V}^\top \tag{5.11}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of $\mathbf{X}^\top\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, see Appendix 5.8.1. Variable projection takes advantage of this closed form solution. Partially minimizing in $\mathbf{A}$ via the SVD has additional advantages over using an iterative algorithm when $\mathbf{A}$ is ill-conditioned. This is an important consideration for robust penalties, where a closed form solution for $\mathbf{A}$ is not immediately available, see Section 5.4.3.

The SPCA problem (5.9) is nonconvex, and so is the value function $v(\mathbf{B})$. To better understand $v(\mathbf{B})$, we consider the following simple 2D example

$$f(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2}\|\mathbf{X} - \mathbf{X}\boldsymbol{b}\boldsymbol{a}^\top\|_{\mathrm{F}}^2 \quad \text{subject to} \quad \mathbf{a}^\top\mathbf{a} = \mathbf{1},$$

where $\mathbf{X} \in \mathcal{R}^{2\times 2}$, $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{R}^2$. We write $v(\mathbf{b}) : \mathbb{R}^2 \to \mathbb{R}$ explicitly as

$$v(\boldsymbol{b}) = \frac{1}{2}\|\mathbf{X} - \mathbf{X}\boldsymbol{b}\,\boldsymbol{a}(\boldsymbol{b})^\top\|_{\mathrm{F}}^2, \quad \boldsymbol{a}(\boldsymbol{b}) = \frac{\mathbf{X}^\top\mathbf{X}\boldsymbol{b}}{\|\mathbf{X}^\top\mathbf{X}\boldsymbol{b}\|}.$$

Figure 5.2 shows the level sets of this function, which are clearly nonconvex. We also see that $v(\mathbf{b})$ is smooth except at $\mathbf{b} = 0$.

We apply proximal gradient methods (see e.g. [275]) to find a stationary point of the value function $v(\mathbf{b})$ (5.10). It is easy to both evaluate $v(\mathbf{b})$ and to compute the gradient. We obtain $\mathbf{a}(\mathbf{b})$ using (5.11) and then use the formula

$$\nabla v(\mathbf{b}) = \nabla_{\mathbf{b}} f(\mathbf{a}, \mathbf{b})|_{\mathbf{a}=\mathbf{a}(\mathbf{b})} = \mathbf{X}^\top(\mathbf{X} - \mathbf{X}\boldsymbol{b}\,\boldsymbol{a}(\boldsymbol{b})^\top)\boldsymbol{a}(\boldsymbol{b}).$$

This yields a simple and efficient algorithm detailed in Algorithm 12. The following theorem provides a sublinear convergence guarantee for Algorithm 12.

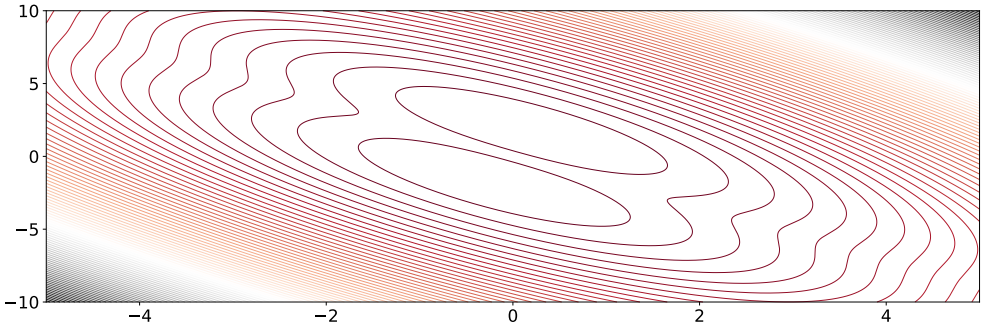


Figure 5.2: Level set of simple 2D projected function.

Theorem 15 (Convergence of 12)**.** *Assume $\rho = \rho_{\mathrm{F}}$, then the optimality criterion satisfies*

$$\min_{1\le k\le N} T(\mathbf{A}_k, \mathbf{B}_k) \le \frac{2(\|\mathbf{X}\|_2^2 + L)^2}{N\|\mathbf{X}\|_2^2} f(\mathbf{A}_1, \mathbf{B}_1),$$

where T is the stationarity of the objective (defined in Appendix 5.8.2) and L is the Lipchitz constant for $\partial\psi$.

See Appendix 5.8.2 for the proof.

Algorithm 12 Variable projected proximal gradient method for (5.10)

Input: $\mathbf{A}_0$, $\mathbf{B}_0$, $k = 0$, $\epsilon > 0$, $\gamma = 1/\|\mathbf{X}\|_2^2$

1: **while** $T(\mathbf{A}_k, \mathbf{B}_k) \ge \epsilon$ **do** ▷ See (5.29)

2: $\quad \mathbf{B}_{k+1} \leftarrow \mathrm{prox}_{\gamma r}(\mathbf{B}_k - \gamma\mathbf{X}^\top(\mathbf{X}\mathbf{B}_k - \mathbf{X}\mathbf{A}_k))$ ▷ See (5.28)

3: $\quad \mathbf{A}_{k+1} \leftarrow \mathrm{argmin}_{\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}^\top\|_{\mathrm{F}}^2 \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}$ ▷ See (5.11)

4: $\quad k \leftarrow k + 1$

5: **end while**

Output: $\mathbf{A}_{k+1}$, $\mathbf{B}_{k+1}$

5.4.2 Randomized Sparse PCA

Randomization allows efficient computation of low-rank approximations such as the SVD and PCA [180, 239, 119, 138]. We form a low-dimensional sketch (representation) of the data, which aims to capture the essential information of the original data. Using this idea, we can reformulate (5.9) as

$$v(\mathbf{B}) := \min_{\mathbf{A}} \frac{1}{2}\|\widetilde{\mathbf{X}} - \widetilde{\mathbf{X}}\mathbf{B}\mathbf{A}^\top\|_F^2 \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \tag{5.12}$$

where $\widetilde{\mathbf{X}} \in \mathbb{R}^{l\times p}$ denotes the sketch of $\mathbf{X} \in \mathbb{R}^{n\times p}$. Here, the dimension l is chosen slightly larger than the target-rank k. We proceed by forming a sample matrix $\mathbf{Y} \in \mathbb{R}^{n\times l}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{\Omega}, \tag{5.13}$$

where $\mathbf{\Omega} \in \mathbb{R}^{p\times l}$ is a randomly generated test matrix [180]. Next, an orthonormal basis matrix is obtained by computing the QR-decomposition of the samples matrix $\mathbf{Y} = \mathbf{Q}\mathbf{R}$. Finally, the sketch is formed by projecting the input matrix to the range of $\mathbf{Y}$, which is low-dimensional:

$$\widetilde{\mathbf{X}} = \mathbf{Q}^\top\mathbf{X}. \tag{5.14}$$

This approach is suitable for input matrices with low-rank structure. The computational advantage becomes significant when the intrinsic rank of the data is relatively small compared to the dimension of the ambient measurement space. The quality of the sketch can be improved by computing additional power iterations [180, 138], especially if the singular value spectrum of $\mathbf{X}$ is only slowly decaying. We suggest computing at least two power iterations by default.

5.4.3 Robust Sparse PCA via Variable Projection

Classically, SPCA is formulated as a least-squares problem, however, it is well-known that the squared loss is sensitive to outliers. In many real world situations we face the challenge that data are grossly corrupted due to measurement errors or other effects. This motivates the need of robust methods which can more effectively account for corrupt or missing data. Indeed, several authors have proposed a robust formulation of SPCA, using the ℓ_1 norm as a robust loss function, to deal with grossly corrupted data [251, 102, 196].

For a robust formulation of SPCA, we use a closely related idea of separating a data matrix into a low-rank model and a sparse model. The architecture is depicted in Figure 5.3. This form of additive decomposition is well-known as robust principal component analysis (RPCA), and its remarkable ability to separate high-dimensional matrices into low-rank and sparse component makes RPCA an invaluable tool for data science [79, 53, 9]. Specifically, we suggest to use the Huber loss function $\rho = \rho_{\mathrm{H}}$ rather than the ℓ_1 norm as the data misfit. The Huber norm overcomes some of the shortcomings for the Frobenius norm and can be

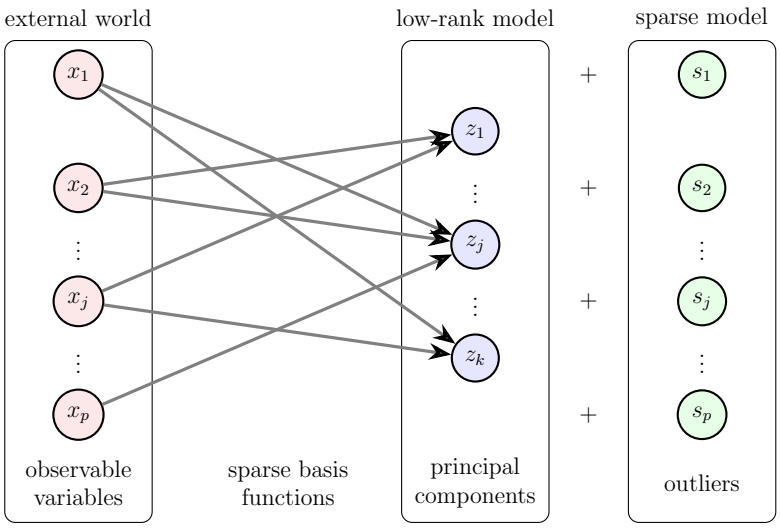


Figure 5.3: Robust SPCA combines a low-rank and sparse model to represent the observable variables. The low-rank model forms the principal components as a sparsely weighted linear combination of the observed variables. The sparse model captures outliers in the data.

used as a more robust measure of fit [195, 245]. We define the Huber loss function as

$$\rho_{\mathrm{H}}(x;\kappa) = \begin{cases} \kappa|x| - \kappa^2/2, & |x| > \kappa \\ x^2/2, & |x| \le \kappa \end{cases},$$
$$\rho_{\mathrm{H}}(\mathbf{A};\kappa) = \sum_{i,j} \rho_{\mathrm{H}}(\mathbf{A}_{ij};\kappa).$$

Figure 5.4 illustrates the least squares and the Huber loss functions. The Huber loss functions grows at a linear rate for residuals outside the thresholding parameter κ, rather than quadratically. Hence, the influence of large deviations on the parameters is reduced. This is consistent with using a heavy tail distribution to model measurement errors.

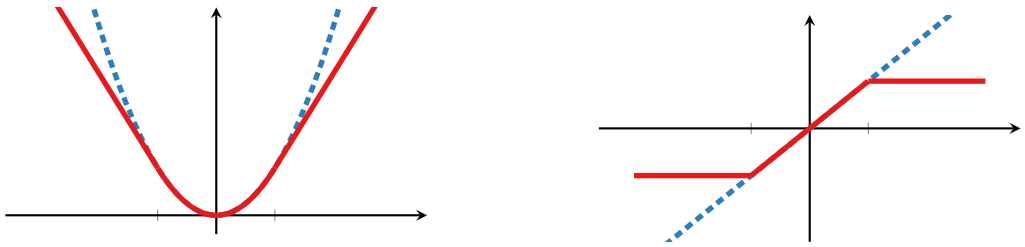
The Huber penalty can be characterized as the (scaled) Moreau envelope of the ℓ_1 norm, see Section 5.8.2:

$$\rho_{\mathrm{H}}(x;\kappa) = \min_{s} \frac{1}{2}\|s - x\|^2 + \kappa\|s\|_1. \tag{5.15}$$

This characterization explicitly brings out outliers s as sparse perturbations to the data. It also makes it possible to develop efficient algorithms for the robust case. In general, our approach applies to any robust norm that can be characterized as the Moreau envelope of a separable penalty.

A naive approach loses the closed form of $\mathbf{A}$ (5.11). To preserve the advantages of partial minimization, we must place the Huber loss on the Procrustean bed of the orthogonal Procrustes problem. We use the Moreau characterization (5.15) to explicitly model sparse outliers using the variable $\mathbf{S}$, and rewrite Eq. (5.9) as follows:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{S}} \quad f_{\mathrm{H}}(\mathbf{A},\mathbf{B},\mathbf{S}) := \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top - \mathbf{S}\|_{\mathrm{F}}^2 + \psi(\mathbf{B}) + \kappa\|\mathbf{S}\|_1 \quad \text{subject to} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \tag{5.16}$$



(a) Loss functions. (b) Influence functions (first derivatives).

Figure 5.4: Illustration of the least-squares loss (dashed blue) and Huber (solid red) loss functions in (a); the first derivatives in (b) can be viewed as influence functions of the residuals.

Now we can again use the orthogonal Procrustes approach (5.11) and reduce (5.16) to minimizing the value function

$$\min_{\mathbf{B},\mathbf{S}} \quad v_{\mathrm{H}}(\mathbf{B},\mathbf{S}) := \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}(\mathbf{B},\mathbf{S})^\top - \mathbf{S}\|_{\mathrm{F}}^2 + \psi(\mathbf{B}) + \kappa\|\mathbf{S}\|_1, \tag{5.17}$$

where $\mathbf{A}(\mathbf{B},\mathbf{S})$ is given by $\mathbf{U}\mathbf{V}^\top$ with

$$(\mathbf{X} - \mathbf{S})^\top \mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top. \tag{5.18}$$

Problem (5.17) has the same structure as (5.10) in the variables $(\mathbf{B},\mathbf{S})$, and we can easily modify the algorithm to account for the additional block, as detailed in Algorithm 13. The partial minimization of $\mathbf{S}$ is a prox evaluation of the 1-norm, which is the soft thresholding operator, see Table 5.1.

Algorithm 13 Gauss-Seidel proximal gradient method for (5.16)

Input: $\mathbf{A}_0, \mathbf{B}_0, \mathbf{S}_0, k = 0, \gamma = 1/\|\mathbf{X}\|_2^2$

1: **while** not converged **do**
2: $\quad \mathbf{B}_{k+1} \leftarrow \mathrm{prox}_{\gamma r}(\mathbf{B}_k - \gamma\mathbf{X}^\top(\mathbf{X}\mathbf{B}_k - \mathbf{X}\mathbf{A}_k + \mathbf{S}_k\mathbf{A}_k))$ ▷ See (5.17)
3: $\quad \mathbf{A}_{k+1} \leftarrow \mathrm{argmin}_{\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}^\top - \mathbf{S}_k\|_{\mathrm{F}}^2$ subject to $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$ ▷ See (5.18)
4: $\quad \mathbf{S}_{k+1} \leftarrow \mathrm{argmin}_{\mathbf{S}} \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}_{k+1}^\top - \mathbf{S}\|_{\mathrm{F}}^2 + \kappa\|\mathbf{S}\|_1$ ▷ See (5.17)
5: $\quad k \leftarrow k + 1$
6: **end while**

Output: $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{S}_k$

5.5 Spatiotemporal SPCA

Sparse decompositions are becoming increasingly relevant for data-driven spatiotemporal analysis of physical systems. The recent proliferation of machine learning and manifold learning methods seek interpretable models using physically meaningful constraints [272, 304, 202, 325, 232]. However, standard orthogonal decompositions such as SVD or proper orthogonal decomposition (POD) may suffer from overfitting and the resulting spatial modes are spatially dense. By promoting sparsity in the modes, SPCA is able to yield modes that may be more interpretable.

The goal of spatiotemporal modal analysis is a system decomposition that is separable in space and time,

$$\mathbf{x}(t) = \sum_{j=1}^{r} a_j(t)\boldsymbol{\phi}_j, \tag{5.19}$$

where $\boldsymbol{\phi}_j$ is a mode evaluated at a grid of spatial locations. Classical data-driven analysis seeks a low-rank approximation given a data matrix of snapshots in time

$$\mathbf{X} = [\mathbf{x}(t_1)\ \mathbf{x}(t_2)\ \ldots\ \mathbf{x}(t_p)]. \tag{5.20}$$

The proper orthogonal decomposition is a canonical data-driven decomposition in the analysis of high-dimensional flows, which seeks the optimal rank-r orthogonal projection of the data that approximates the covariance of $\mathbf{X}$. The optimal low-rank projection is given by the dominant k scaled principal components $\mathbf{Z} = \mathbf{XV} = \mathbf{U\Sigma}$, obtained from the singular value decomposition $\mathbf{X} = \mathbf{U\Sigma V}^T$. We define the modes to be $\boldsymbol{\Phi} = \mathbf{U}$, resulting in the separable decomposition

$$\mathbf{X} = \boldsymbol{\Phi}\mathbf{C}^{\mathbf{T}}, \tag{5.21}$$

where $\mathbf{C}^{\mathbf{T}} = \boldsymbol{\Sigma}\mathbf{V}^T$. While POD modes numerically approximate the data, they may not be physically meaningful. POD modes do not generally correspond to coherent structures that persist in time. Sparse PCA, on the other hand, imposes sparsity in the spatial modes while maintaining time independence. In our framework, spatial modes are given by

$$\boldsymbol{\Phi} = \mathbf{XB} = \begin{bmatrix} | & | & & | \\ \mathbf{Xb}_1 & \mathbf{Xb}_2 & \ldots & \mathbf{Xb}_k \\ | & | & & | \end{bmatrix}, \tag{5.22}$$

which represents a sparse linear combination of the snapshots and sparse modes $\boldsymbol{\Phi}$. Recall, the columns of $\mathbf{B}$ are the sparse weight vectors. As we shall demonstrate, SPCA modes display greater correspondence to coherent structures in various flows.

5.6 Results

We now apply our SPCA framework to a number of example systems of interest, ordered by increasing complexity. These examples capture many challenges that motivate the new algorithms. The first example is an artificial dataset with high-dimensional measurements and low-dimensional structures across multiple scales. In this example, there is a ground truth, providing a straightforward benchmark for SPCA and robust SPCA. The second example applies SPCA to a highly structured fluid flow, characterized by laminar vortex shedding behind a circular cylinder at low Reynolds number, which is a benchmark problem in fluid dynamics [261]. Fluid flows are ideal for developing interpretable models of multi-scale physics and deploying sparse sensors for estimation and control. This is because they are high-dimensional systems that often exhibit low-dimensional coherent patterns that are spatially localized [60, 323]. The third example involves high-dimensional satellite data of the ocean surface temperature, a complex multiscale system that is intimately related to global circulation and climate. In all of these examples, the data is dynamic, high-dimensional, exhibits low-dimensional patterns at multiple scales, and has fewer snapshots in time than measurements in space. The proposed SPCA framework allows efficient computations on large systems, yields robust estimates from noisy data, and gives interpretable modes that can be used for the downstream tasks in dynamical systems modeling and control.

5.6.1 Multiscale Video Example

First, we consider a case where spatiotemporal dynamics are generated from three spatial modes oscillating at different frequencies in overlapping time intervals:

$$\mathbf{x}(t) = \sum_{j=1}^{3} a_k(t)\boldsymbol{\phi}_j. \tag{5.23}$$

The multiscale time dynamics switch on and off irregularly, i.e., the modes effectively appear mixed in time as is common in other real-world phenomena such as weather, climate, etc. Consequently, within a single frame, the three modes occasionally mix, rendering the disambiguation task more challenging. However, we see that SPCA is able to recover the three modes in an unsupervised manner. Specifically, the data is generated on a 200×200 spatial grid for 150 seconds with timestep $\Delta t = .5$s. We flatten the spatial dimensions to obtain a data matrix with $p = 40,000$ measurements for each of the $n = 300$ snapshots (observations in time).

The results of PCA and SPCA on the raw frame data are compared and contrasted in Fig. 5.6. Here the spatial coherent structures extracted by SPCA recover the generating spatiotemporal modes, while PCA is unable to do so. By seeking a parsimonious representation, SPCA is able to accurately correlate spatial structures with their individual time histories. Because PCA has no such constraint, the different spatial structures remain mixed.

In many applications data exhibit grossly corrupted entries that typically arise from process or measurement noise. The least-squares loss function is sensitive to outliers. Thus, SPCA tends to be biased and the results can be misleading. To overcome this, our proposed robust SPCA algorithm can be used. The Huber loss function allows one to separate the input data into a low-rank component plus a sparse component. This is demonstrated in Fig. 5.7. The robust implementation clearly separates the polluted data into a low-rank component, while capturing the additive salt and pepper noise. However, the robust implementation is computationally more demanding than the standard SPCA algorithm.

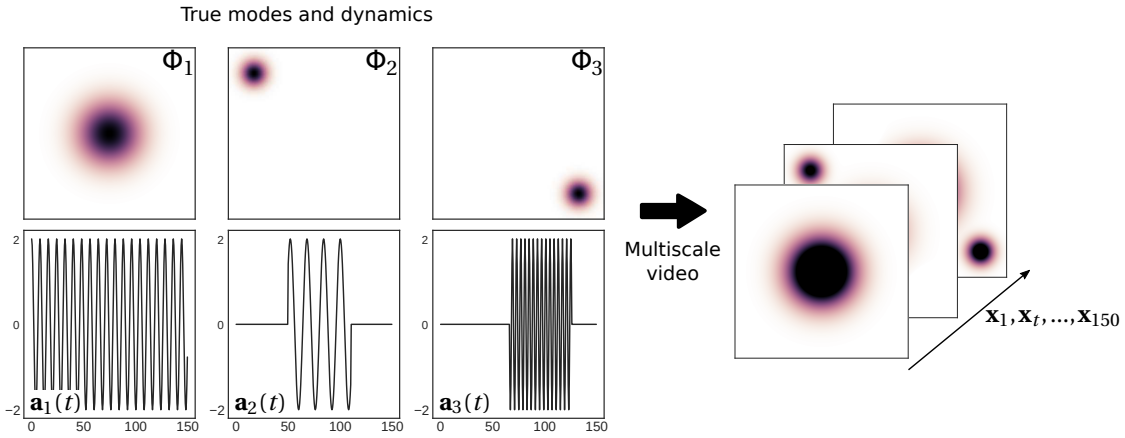
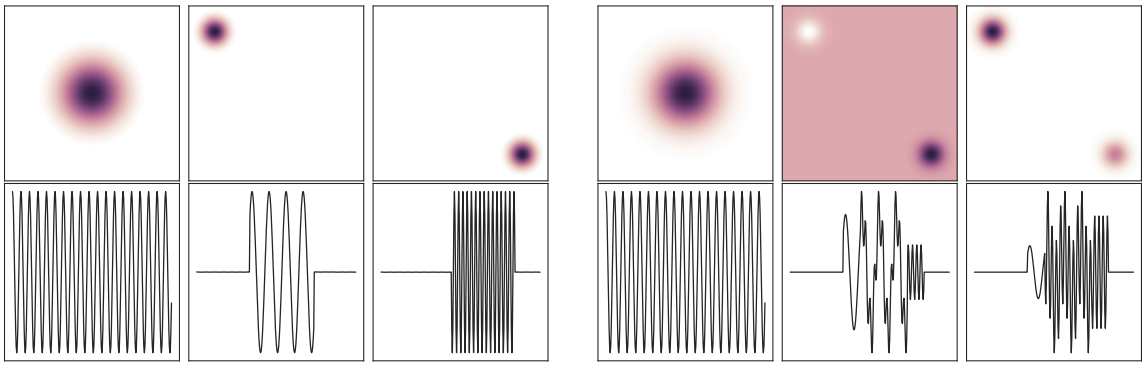
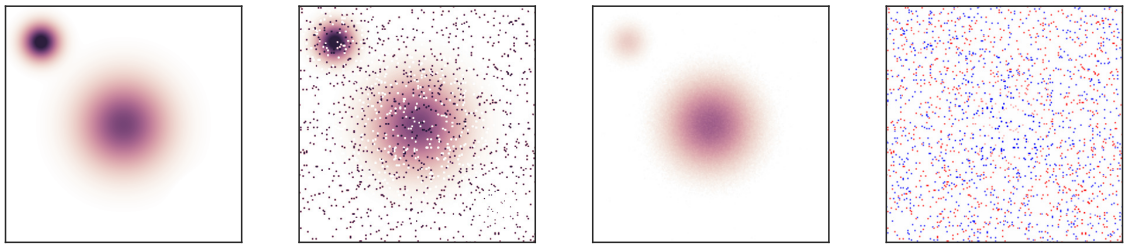


Figure 5.5: Multiscale video model. Each frame of this multiscale video is high-dimensional with 200×200 pixels, however the system has only three degrees of freedom.



(a) SPCA. (b) PCA.

Figure 5.6: Multiscale video reconstruction. SPCA successfully decomposes the video into the true dynamics, while PCA fails to disambiguate modes 2 and 3.

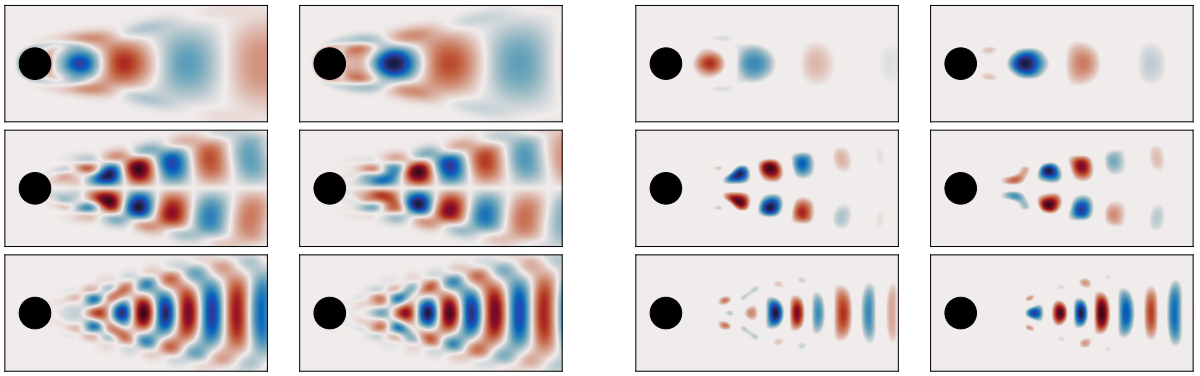


(a) Truth. (b) Corrupted. (c) Reconstruction. (d) Outliers.

Figure 5.7: Approximation of a grossly corrupted multiscale video using robust SPCA. Here the low-rank approximation with robust PCA (c) successfully recovers the true frame and filters out added salt and pepper noise (d).

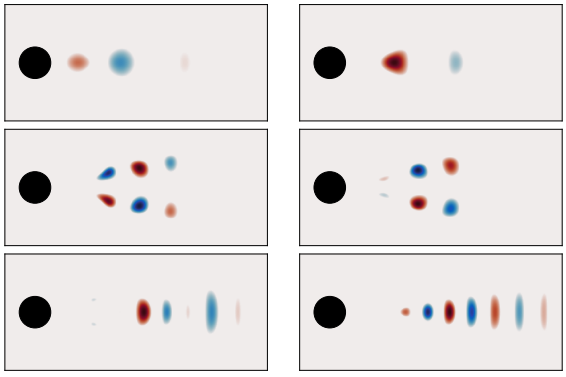
5.6.2 Fluid Flow

PCA has been extensively used in fluid dynamics for decades, where it is known as proper orthogonal decomposition, providing a data-driven generalization of the Fourier transform [41]. Here we apply SPCA to the flow behind a cylinder, a canonical example in fluid dynamics [261]. The data consists of a time series of the vorticity field behind a solid cylinder at Reynolds number 100, which induces laminar vortex shedding downstream. The flow is simulated using an immersed boundary projection method [324] on a 450×200 spatial grid for 3 dimensionless time units with timestep $\Delta t = .02$. Again, we flatten the spatial dimensions and yield a data matrix with $p = 90,000$ measurements for each of the $n = 150$ snapshots (observations in time). The resulting principal components or spatial modes of the flow are

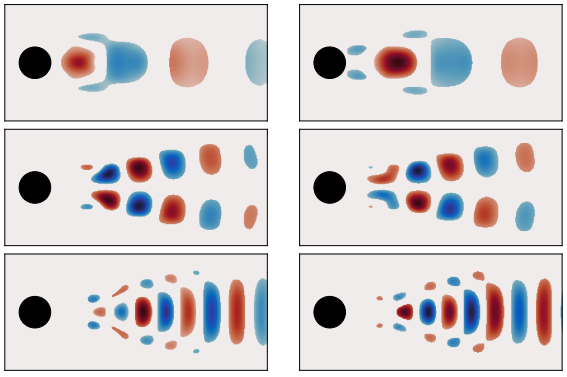


(a) PCA.

(b) SPCA with ℓ_1 regularization ($\alpha = 10^{-5}$).



(c) SPCA with ℓ_1 regularization ($\alpha = 10^{-4}$).



(d) SPCA with ℓ_0 regularization ($\alpha = 10^{-5}$).

Figure 5.8: Sparse PCA demonstrates superior separation of the spatial eigenmodes responsible for vortex shedding. As a result, we can better differentiate their spatial influence on different regions of the flow downstream of the cylinder.

widely used for reduced-order modeling, prediction, and control.

The SPCA and PCA eigenmodes are compared in Fig. 5.8. Both decompositions successfully identify the dominant mode pairs that occur at characteristic harmonic frequencies. However, the mode structures extracted by SPCA are well-bounded and more interpretable, resulting in visible weakening downstream and stronger influence upstream. This is typical of the vortex shedding regime as vortices dissipate while advecting downstream and is not observed in the PCA modes.

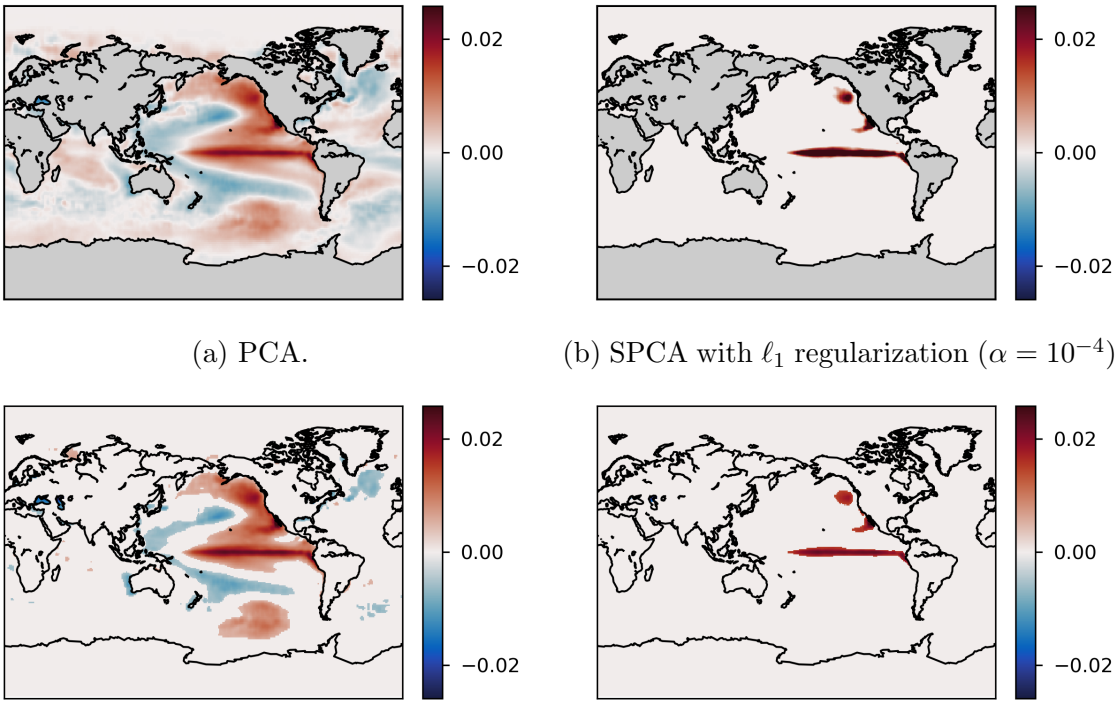
Standard PCA has beneficial orthonormality properties that are crucial for projection based reduced-order modeling of high-dimensional systems. However, as experiments and models simulate increasingly complex flows, the field is rapidly moving towards more interpretable decompositions for learning and control. Recent directions in network analysis of turbulence and mixing require robust tracking of sparse spatial structures and vortices. The ability of SPCA to delineate boundaries of vortex dynamics are critical for the scalable decomposition of such high-resolution flow data. Furthermore, SPCA is purely data-driven and works equally well for modal decomposition of high-fidelity computational fluid dynamics (CFD) simulation, as well as robust denoising of experimental data generated by particle image velocimetry and other high-resolution imaging techniques.

5.6.3 NOAA Ocean Surface Temperature

We now apply SPCA to satellite ocean temperature data from 1990-2017 [292], and compare SPCA results from PCA.[1] The data consists of $n = 1,458$ temporal snapshots which measure the weekly temperature means at $360 \times 180 = 64,800$ spatial grid points. Since we omit data over continents and land, the ambient dimension reduces to $p = 44,219$ observations in our analysis. Our objective is the accurate identification of the intermittent El Niño and La Niña warming events, which are famously implicated in global weather patterns and climate change. The El Niño Southern Oscillation (ENSO) is defined as any sustained temperature anomaly above running mean temperature with a duration of 9 to 24 months. In climate sciences, principal components are also known as empirical orthogonal functions or EOFs; however, traditional PCA struggles to find a low-rank representation of this complex, high-dimensional system.

The canonical El Niño is associated with a narrow band of warm water off coastal Peru that is commonly referred as NIÑO 1+2, 3, 3.4, or 4 to differentiate the types of bands. Traditional PCA is unable to isolate this band, instead combining it with broader spatial signatures across the Pacific and Atlantic in mode 4 (Fig. 5.9a). Nevertheless, this mode is often used to compute the canonical Oceanic Niño Index (ONI). On the other hand, SPCA obtains a dramatic and clean separation of NIÑO 1-4 within the 4th mode (Fig. 5.9b). This is contextualized by the associated temporal mode, which yields sharper peaks during the 1997-1999 and 2014-2016 major El Niño events compared to PCA. The 12-month moving

[1]The data are provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA, and are accessible via their Web site at `https://www.esrl.noaa.gov/psd/`.



(a) PCA.

(b) SPCA with ℓ_1 regularization ($\alpha = 10^{-4}$).

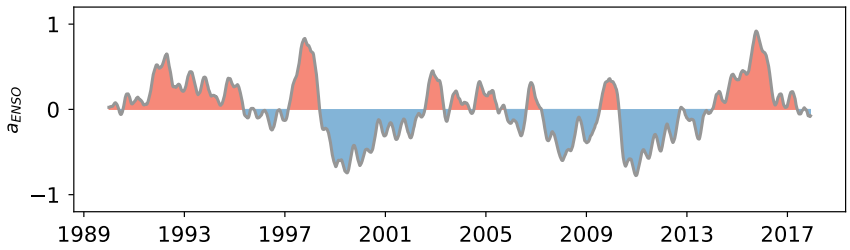
(c) SPCA with ℓ_0 regularization ($\alpha = 10^{-5}$).

(d) SPCA with ℓ_0 regularization ($\alpha = 10^{-4}$).

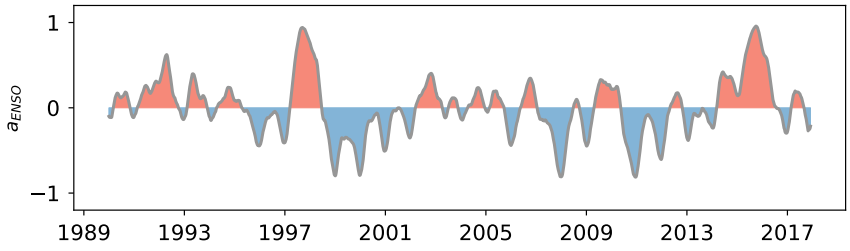
Figure 5.9: SPCA successfully identifies the band of warmer temperatures (4th mode) in the South Pacific traditionally associated with El Niño. By contrast, the corresponding PCA mode (4th mode) picks up spurious spatial correlations across the globe.

average of the temporal modes for both PCA and SPCA is shown in Fig. 5.10 and confirms that SPCA differentiates major and minor ENSO events with greater clarity than PCA.

Previous study of this dataset has required a multiresolution time-frequency separation of the data matrix in order to clearly identify the ENSO mode in an unsupervised manner [211].Without the sparsity constraint, SVD-based methods struggle to obtain a low-rank representation of these complex systems with nonlinear dynamics, coupled interactions and multiple timescales of motion. Based on our findings, SPCA has the potential to yield sparse modal representations of complex systems and coherent structures that may alter our understanding of oceanic and atmospheric phenomena.



(a) PCA modes.



(b) SPCA with ℓ_1 regularization ($\alpha = 10^{-4}$).

Figure 5.10: Oceanic Niño Index (ONI), a 12-month moving average of the ENSO mode, reveals greater distinction between major (1997-1999,2014-2016) and minor events with SPCA modes.

5.6.4 Denoising with Sparse PCA

Cumulative variance plots reveal that sparse PCA behaves differently on the latter two examples. This can be attributed to the level of stochasticity in each system. The cylinder data has high temporal resolution and is therefore sufficiently well-resolved for sparse PCA to capture nearly all the variance within the low-rank component (Fig. 5.11a). In this case the decomposition is similar to PCA, although spatially more localized. On the other hand, the ocean data has coarse weekly temporal resolution. Therefore, faster dynamics which are not sufficiently resolved appear stochastic. Hence, slower timescales (annual, ENSO) are reflected in the low-rank component of SPCA as indicated by cumulative variance (Fig. 5.11b). PCA, however, overfits with 'noisy' components which are not physically meaningful.

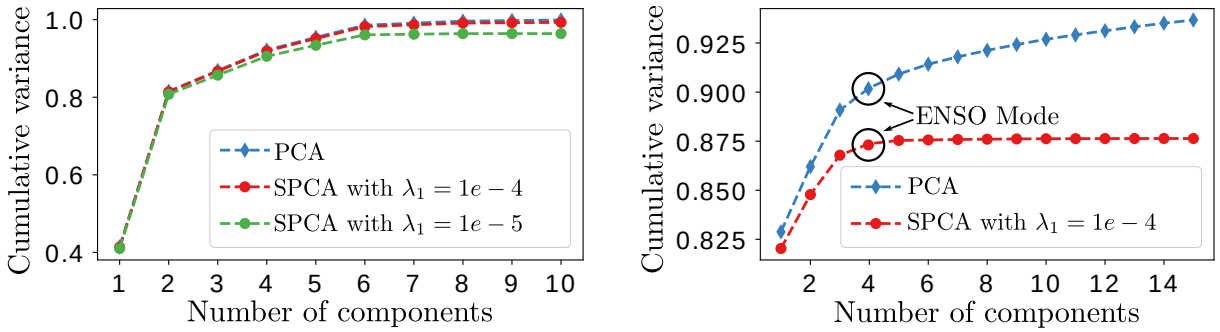
5.6.5 Computational Performance

To demonstrate the computational performance of the proposed SPCA algorithms we compute the leading $k = 10$ components for two data matrices. First, we consider the cases of small p data. Figure 5.12 shows the time until the objective function converges within a tolerance level of 10^{-5}. For comparison we show the performance of the SPCA algorithm using least angle regression (LARS) and coordinate descent (CD) as proposed by [387]. Our

proposed algorithm based on variable projection outperforms both the LARS and CD algorithm.

Further, the randomized accelerated SPCA algorithm outperforms the deterministic variable projection algorithms. The desired accuracy is achieved about 5 times faster compared to the deterministic algorithm. This is despite the fact that the randomized algorithms require more iterations than the deterministic algorithm to converge. The computational advantage is even greater for the high-dimensional data setting (i.e., big p) as shown in Figure 5.13 The computational advantage of the randomized algorithm becomes pronounced with increasing dimensions of the input matrix. Hence, the randomized algorithm allows exploring a large space of tuning parameters and is well suited for performing cross-validation.

Implementations of our algorithms are provided in Python `https://github.com/erichson/ristretto` and in R `https://CRAN.R-project.org/package=sparsepca`.



(a) Although sparsity promotes spatially localized structure, SPCA retains nearly all the variance of the fluid flow system.

(b) SPCA clearly isolates the fourth ENSO mode as the last physically relevant component to the system.

Figure 5.11: Cumulative variance of each component. SPCA approximates PCA to varying degrees for the two fluid datasets. In contrast to PCA, SPCA separates the ENSO mode from noisy contributions even though ENSO captures only 1% of the total variance.

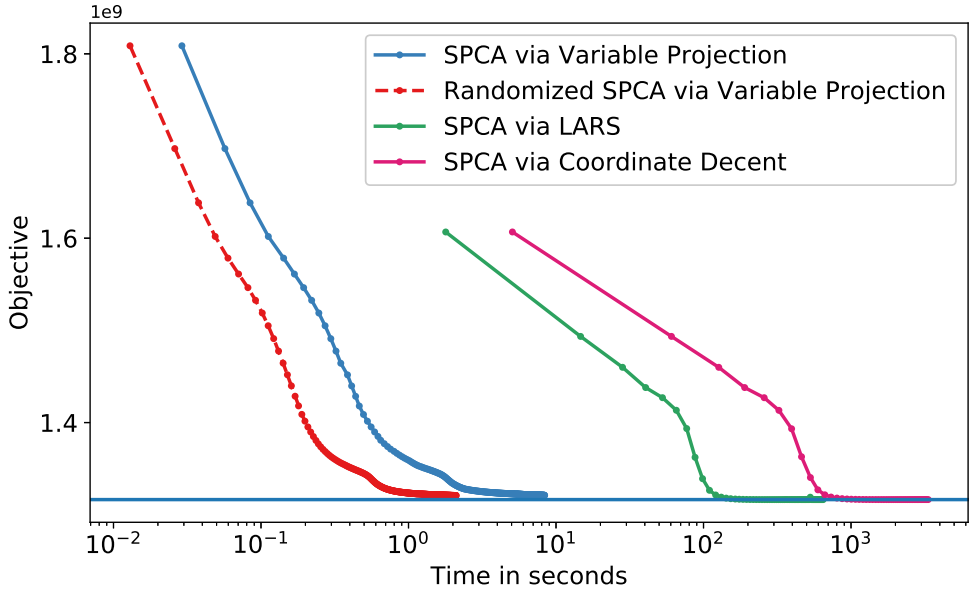


Figure 5.12: Computational performance of different SPCA algorithm. The dominant 10 sparse weight vectors are computed for a 2000×1344 data matrix.

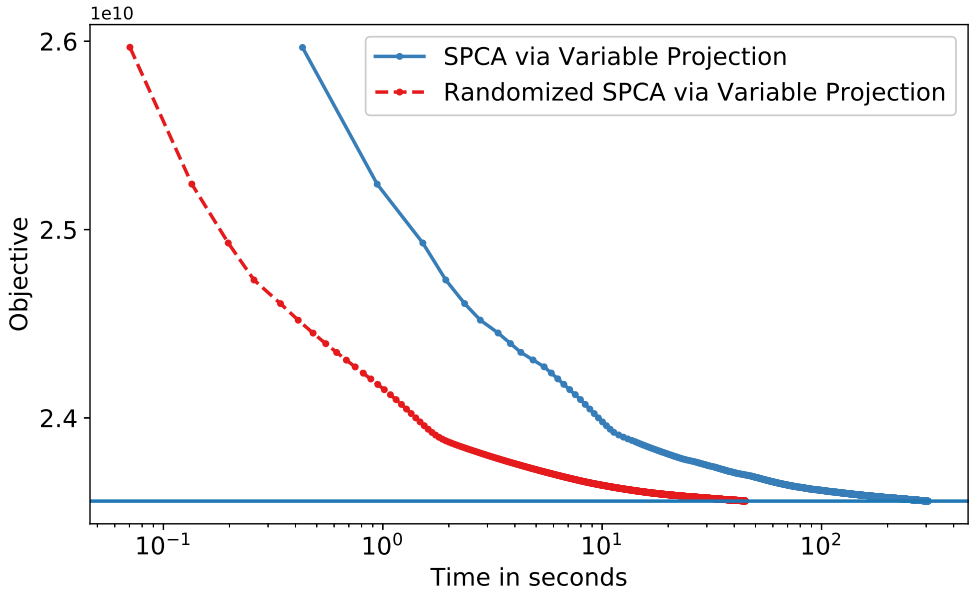


Figure 5.13: Computational performance of the randomized and deterministic SPCA algorithm using variable projection. The dominant 10 sparse weight vectors are computed for a 2000×16128 data matrix. The randomized algorithms is about 4 times faster.

5.7 Discussion

We have presented a robust and scalable architecture for computing sparse principal component analysis (SPCA). Specifically, we have modeled SPCA as a matrix factorization problem with orthogonality constraints, and developed specialized optimization algorithms that partially minimize a subset of the variables (variable projection). Our SPCA algorithm is scalable and robust, greatly improving computational efficiency over current state-of-the-art methods while retaining comparable performance. More precisely, we have demonstrated that: (i) The value function view approach provides an efficient and flexible framework for SPCA; (ii) Robust SPCA can be formulated using the Huber loss; (iii) A wide variety of sparsity-inducing regularizers can be incorporated into the framework; (iv) The proposed algorithms are computationally efficient for high-dimensional data, i.e, large p; (v) Randomized methods for linear algebra substantially eases the computational demands, while obtaining a near-optimal approximation for low-rank data.

SPCA is a useful diagnostic tool for data featuring rich dynamics that give rise to multiscale structures in both space and time. Given that such phenomena are ubiquitous in the physical, engineering, biological, and social sciences, this work provides a valuable tool for improved interpretability, especially in the diagnostics of localized structures and disambiguation of distinct time scale physical processes. The work also opens a number of avenues for future development:

Methodological Extensions This scalable approach for identifying spatially localized spatial structures in high-dimensional and multiscale data may be directly applied to 1) tensor decompositions [97, 57, 136], which represent data in a multi-dimensional array structure, 2) parsimonious dynamical systems models [61], which identify the fewest nonlinear interactions required to capture the underlying physical mechanisms, and 3) *in situ* sensing and control, where sensors and actuators are generally required to be spatially localized [242].

Applications in the Engineering and Physical Sciences. The methods developed here will be broadly applicable to dynamical systems that are high-dimensional, multiscale, and where there is a need for interpretable and parsimonious models for prediction, estimation, and control. Specific applications where SPCA has already been applied include genomics, and biological systems in general, and atmospheric chemistry. In addition, there is tremendous opportunity for advances in diverse fields, such as improving climate science, detecting and controlling structures in the brain, and closed loop control of turbulent fluid systems [60].

5.8 Appendix

5.8.1 The Orthogonal Procrustes Problem

We seek an orthonormal matrix $\mathbf{A}$ so that

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^\top\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \mathbf{A}^\top\mathbf{A} = \mathbf{I}. \tag{5.24}$$

Indeed, a closed form solution is provided by the SVD. First,we expand the above objective function as

$$\underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}\|_F^2 + \|\mathbf{X}\mathbf{B}\|_F^2 - 2 \cdot \operatorname{trace}(\mathbf{X}^\top\mathbf{X}\mathbf{B}\mathbf{A}^\top).$$

This problem is equivalent to find a orthonormal matrix $\mathbf{A}$ which maximizes $\operatorname{trace}(\mathbf{X}^\top\mathbf{X}\mathbf{B}\mathbf{A}^\top)$. We proceed by substituting the SVD of $\mathbf{X}^\top\mathbf{X}\mathbf{B}$ so that we yield

$$\underset{\mathbf{A}}{\operatorname{argmax}} \operatorname{trace}(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{A}^\top) = \operatorname{trace}(\mathbf{\Sigma}\mathbf{V}^\top\mathbf{A}^\top\mathbf{U}). \tag{5.25}$$

Note that $\mathbf{\Sigma}$ is a diagonal matrix with non-negative entries and $\mathbf{V}^\top\mathbf{A}^\top\mathbf{U}$ is an orthonormal matrix for any matrix $\mathbf{A}^\top$. Because of this, the trace norm in Eq. (5.25) is maximized by $\mathbf{A}^\top$ which turns $\mathbf{V}^\top\mathbf{A}^\top\mathbf{U}$ into an identity matrix $\mathbf{I}$, so that we yield $\operatorname{trace}(\mathbf{\Sigma}\mathbf{I})$. Hence, an optimal solution is provided by $\mathbf{A} = \mathbf{U}\mathbf{V}^\top$, i.e., the left and right singular vectors of $\mathbf{X}^\top\mathbf{X}\mathbf{B}$.

5.8.2 Proof of Theorem

Technical Preliminaries

In the following we give a brief overview of notation and concepts used to develop and analyze the algorithms in this paper. Further, we review briefly the elements of variational analysis for the theoretical analysis of the algorithm [254, 295].

Matrix Spaces We consider the collection of all matrices with the same dimension $\mathcal{R}^d$ (where d could be shorthand for $p \times p$) as a Hilbert space equipped with the inner product. More concretely, the inner product is defined by the trace and the norm induced by this inner product is the Frobenius norm

$$\langle \mathbf{M} \rangle \, \mathbf{M} := \operatorname{trace}(\mathbf{M}^\top\mathbf{M}) = \|\mathbf{M}\|_{\mathrm{F}}^2.$$

For any map $\mathbf{\Phi} : \mathcal{R}^d \to \mathcal{R}^l$, we set,

$$\operatorname{lip}(\mathbf{\Phi}) := \sup_{\mathbf{M} \neq \mathbf{N}} \frac{\|\mathbf{\Phi}(\mathbf{M}) - \mathbf{\Phi}(\mathbf{N})\|_{\mathrm{F}}}{\|\mathbf{M} - \mathbf{N}\|_{\mathrm{F}}}$$

We say that $\mathbf{\Phi}$ is L-Lipschitz continuous, for some $L \geq 0$, if the inequality $\operatorname{lip}(\mathbf{\Phi}) \leq L$ holds.

Functions and Geometry Constraints, such as those in (5.7), can be represented using functions from a matrix space $\mathbb{R}^d$ to the extended real line defined by $\overline{\mathcal{R}} := \mathcal{R} \cup \{\pm\infty\}$. The *domain* and the *epigraph* of any function $f : \mathcal{R}^d \to \mathcal{R}$ are the defined sets

$$\begin{aligned} \operatorname{dom} f &:= \{\mathbf{M} \in \mathcal{R}^d : f(\mathbf{M}) < +\infty\}, \\ \operatorname{epi} f &:= \{(\mathbf{M}, r) \in \mathcal{R}^d \times \mathcal{R} : f(\mathbf{M}) \leq r\}. \end{aligned}$$

For any set $\mathcal{F} \subset \mathcal{R}^d$, we define the *distance*, *projection* and *indicator function* for $\mathbf{M} \in \mathcal{R}^d$ by

$$\operatorname{dist}(\mathbf{M}; \mathcal{F}) := \inf_{\mathbf{N} \in \mathcal{F}} \|\mathbf{N} - \mathbf{M}\|, \quad \operatorname{proj}(\mathbf{M}; \mathcal{F}) := \operatorname*{argmin}_{\mathbf{N} \in \mathcal{F}} \|\mathbf{N} - \mathbf{M}\|,$$
$$\delta_{\mathcal{F}}(\mathbf{M}) := \begin{cases} 0, & \mathbf{M} \in \mathcal{F} \\ \infty, & \mathbf{M} \notin \mathcal{F} \end{cases}.$$

For $\mathbb{O} := \{\mathbf{A} \in \mathcal{R}^d : \mathbf{A}^T\mathbf{A} = \mathbf{I}\}$ in (5.7), and given $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathbf{T}$, we have

$$\operatorname{dist}(\mathbf{M}; \mathbb{O}) := \|\mathbf{I} - \mathbf{\Sigma}\|^2, \quad \operatorname{proj}(\mathbf{M}; \mathbb{O}) := \mathbf{U}\mathbf{V}^\mathbf{T}, \quad \delta_{\mathbb{O}}(\mathbf{M}) := \begin{cases} 0, & \mathbf{\Sigma} = \mathbf{I} \\ \infty, & \mathbf{\Sigma} \neq \mathbf{I}. \end{cases} \tag{5.26}$$

Subgradients and Subdifferentials Characterizing stationarity (a necessary condition for optimality) is a key step in analyzing the behavior of an algorithm and deriving practical termination criteria. Problem (5.7) is nonsmooth, so gradients do not exist. Instead, we can use more general concepts of *subgradients*, which exist for nonsmooth, nonconvex functions.

Consider an arbitrary function $f \colon \mathcal{R}^d \to \overline{\mathcal{R}}$ and a point $\overline{\mathbf{M}}$ with $f(\overline{\mathbf{M}})$ finite. When f is convex, the subgradient of f at $\overline{\mathbf{M}}$ is defined as the collection of tangent affine minorants:

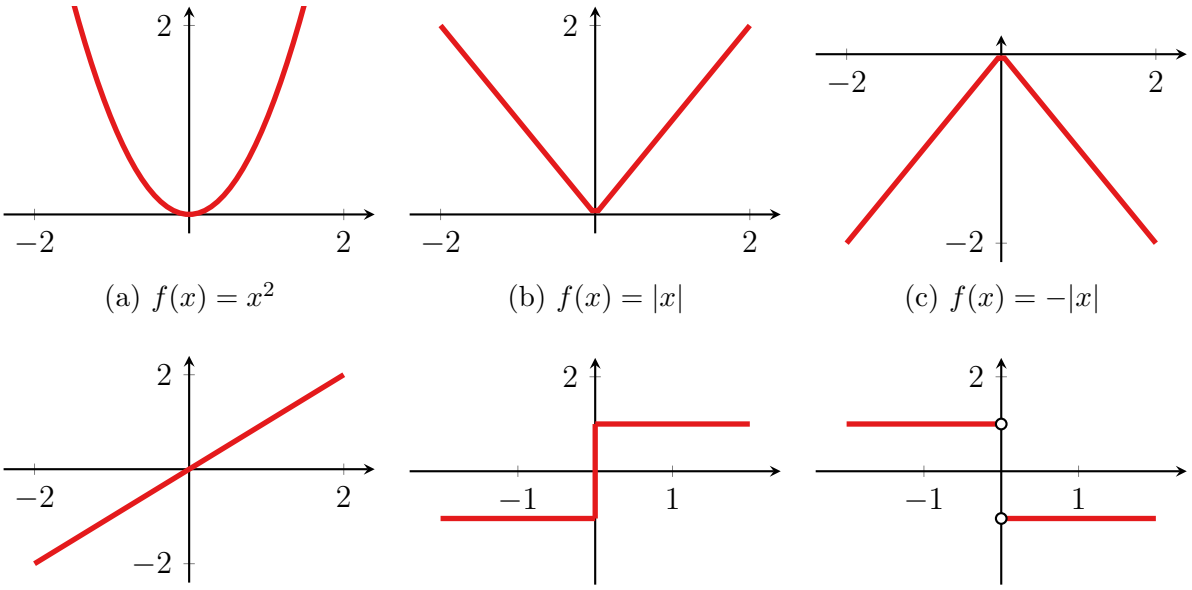
$$\partial f(\overline{\mathbf{M}}) := \{\mathbf{V} : f(\mathbf{M}) \geq f(\overline{\mathbf{M}}) + \langle \mathbf{V}, \mathbf{M} - \overline{\mathbf{M}} \rangle\}. \tag{5.27}$$

If f is differentiable at $\overline{\mathbf{M}}$, then $\partial f(\overline{\mathbf{M}})$ contains only one element, and it is a gradient. When f is not differentiable, the subdifferential can contain multiple elements (see Figure 5.14). From (5.27), it is clear that $0 \in \partial f(\overline{\mathbf{M}})$ implies that $f(\mathbf{M}) \geq f(\overline{\mathbf{M}})$ for all $\mathbf{M}$, i.e. $\overline{\mathbf{M}}$ is a global minimum.

When f is nonconvex, (5.27) may not hold globally for any $\mathbf{V}$, and we need a localized definition. The *Fréchet subdifferential* of f at $\overline{\mathbf{M}}$, denoted $\hat{\partial} f(\overline{\mathbf{M}})$, is the set of all matrices $\mathbf{V}$ that satisfy

$$f(\mathbf{M}) \geq f(\overline{\mathbf{M}}) + \langle \mathbf{V}, \mathbf{M} - \overline{\mathbf{M}} \rangle + o(\|\mathbf{M} - \overline{\mathbf{M}}\|)$$

as $\mathbf{M} \to \overline{\mathbf{M}}$. The inclusion $\mathbf{V} \in \hat{\partial} f(\overline{\mathbf{M}})$ holds precisely when the affine function $\mathbf{M} \mapsto f(\overline{\mathbf{M}}) + \langle \mathbf{V}, \mathbf{M} - \overline{\mathbf{M}} \rangle$ underestimates f up to first-order near $\overline{\mathbf{M}}$. The limit of Fréchet subgradients $v_i \in \hat{\partial} f(\mathbf{M}_i)$ along a sequence $\mathbf{M}_i \to \overline{\mathbf{M}}$ may not be a Fréchet subgradient at the limiting point $\overline{\mathbf{M}}$. The *limiting subdifferential* $\partial f(\overline{\mathbf{M}})$ is the set of all matrices $\mathbf{V}$ for which there



(a) $f(x) = x^2$ (b) $f(x) = |x|$ (c) $f(x) = -|x|$

(d) $\partial(\cdot)^2(0) = \nabla(\cdot)^2(0) = 0$. (e) $\partial|\cdot|(0) = [-1, 1]$. (f) $\partial - |\cdot|(0) = \{-1, 1\}$.

Figure 5.14: Subgradients are illustrated for the following three cases: (a) smooth function $f(x) = x^2$, (b) a nonsmooth function $f(x) = |x|$, (c) a nonsmooth and nonconvex function $f(x) = |x|$. Subplots (d) to (f) show the corresponding subgradients.

exist sequences $\mathbf{M}_i$ and $\mathbf{V}_i$ that satisfy $\mathbf{V}_i \in \partial f(\mathbf{M}_i)$ and $(\mathbf{M}_i, f(\mathbf{M}_i), \mathbf{V}_i) \to (\overline{\mathbf{M}}, f(\overline{\mathbf{M}}), \mathbf{V})$. In the nonconvex case, the stationarity condition $0 \in \partial f(\overline{\mathbf{M}})$ no longer implies global (or local) optimality. However, it is still a necessary condition, and one that can be checked. We characterize stationarity of (5.7) by the distance of 0 to the limiting subdifferential $\partial f(\overline{\mathbf{M}})$.

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

$$\begin{aligned} f_\gamma(\mathbf{M}) &:= \inf_{\mathbf{L}} \left\{ f(\mathbf{L}) + \frac{1}{2\gamma}\|\mathbf{L} - \mathbf{M}\|^2 \right\}, \\ \operatorname{prox}_{\gamma f}(\mathbf{M}) &:= \operatorname*{argmin}_{\mathbf{L}} \left\{ f(\mathbf{L}) + \frac{1}{2\gamma}\|\mathbf{L} - \mathbf{M}\|^2 \right\}. \end{aligned} \tag{5.28}$$

Theorem 16 (Regularization properties of the envelope)**.** *Let $f\colon \mathcal{R}^d \to \mathcal{R}$ be a proper closed convex function. Then f_γ is convex and C^1-smooth with*

$$\nabla f_\gamma(\mathbf{M}) = \tfrac{1}{\gamma}(\mathbf{M} - \operatorname{prox}_{\gamma f}(\mathbf{M})) \quad \text{and} \quad \operatorname{lip}(\nabla f_\gamma) \le \tfrac{1}{\gamma}.$$

Proof. See Theorem 2.26 of [295]. □

Optimality Condition

The objective function in Equation (5.9) is non-convex. Hence, we rely on iterative schemes that can find the stationary points of the objective.

Definition 6 (Stationary Points)**.** *Assume that ρ is smooth, we call a pair $(\mathbf{A}, \mathbf{B})$ a stationary point when it satisfies*

$$\begin{aligned}\mathbf{0} &\in \nabla\rho(\mathbf{A}\mathbf{X}^\top\mathbf{B}^\top - \mathbf{X}^\top)\mathbf{B}\mathbf{X} + \partial\varphi(\mathbf{A}),\\ \mathbf{0} &\in \mathbf{X}^\top\nabla\rho(\mathbf{X}\mathbf{B}\mathbf{A}^\top - \mathbf{X})\mathbf{A} + \partial\psi(\mathbf{B}),\end{aligned}$$

where $\partial\psi$ is the limiting subdifferential defined in Section 5.8.2. We also introduce a measure of non-stationarity T:

$$\begin{aligned}T(\mathbf{A},\mathbf{B}) = \min\{&\tfrac{1}{2}\|\mathbf{U}\|_{\mathrm{F}}^2 + \tfrac{1}{2}\|\mathbf{V}\|_{\mathrm{F}}^2 :\\ &\mathbf{U} \in \nabla\rho(\mathbf{A}\mathbf{X}^\top\mathbf{B}^\top - \mathbf{X}^\top)\mathbf{B}\mathbf{X} + \partial\varphi(\mathbf{A}),\\ &\mathbf{V} \in \mathbf{X}^\top\nabla\rho(\mathbf{X}\mathbf{B}\mathbf{A}^\top - \mathbf{X})\mathbf{A} + \partial\psi(\mathbf{B}).\}\end{aligned} \tag{5.29}$$

In the main text, we show some examples of prox operator corresponding to $\partial\psi(\mathbf{B})$, but for $\partial\varphi(\mathbf{A})$ it might be hard to understand.

Here we give a simple instance of $\partial\varphi(\mathbf{A})$ when $\phi(\mathbf{A}) = \delta_0(\mathbf{A}|\mathbf{A}^\top\mathbf{A} = \mathbf{I}) = \delta_0(\mathbf{A}|\mathbb{O}_2)$, the space of orthonormal matrices. When consider orthogonal matrices in two dimension, we could characterize them by a single angle variable θ,

$$\mathbf{A} = \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{bmatrix}, \text{ if } \det(\mathbf{A}) = 1, \quad \mathbf{A} = \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ \sin(\theta) & -\cos(\theta)\end{bmatrix}, \text{ if } \det(\mathbf{A}) = -1.$$

For every $\mathbf{A}$ described as above, we define the tangent direction in $\mathbb{O}_2$:

$$\mathbf{T}_\mathbf{A} = \begin{bmatrix}-\sin(\theta) & \cos(\theta)\\ -\cos(\theta) & -\sin(\theta)\end{bmatrix}, \text{ if } \det(\mathbf{A}) = 1, \quad \mathbf{T}_\mathbf{A} = \begin{bmatrix}-\sin(\theta) & \cos(\theta)\\ \cos(\theta) & \sin(\theta)\end{bmatrix}, \text{ if } \det(\mathbf{A}) = -1.$$

We now have

$$\partial\varphi(\mathbf{A}) = \{\mathbf{G} : \langle\mathbf{G}\rangle\, \mathbf{T}_\mathbf{A} = 0\}.$$

In particular, for $\det(\mathbf{A}) = 1$ every element in $\partial\varphi(\mathbf{A})$ is a linear combination of the matrices

$$\begin{bmatrix}-\cos(\theta) & 0\\ \sin(\theta) & 0\end{bmatrix}, \quad \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ 0 & 0\end{bmatrix}, \quad \begin{bmatrix}0 & 0\\ \sin(\theta) & -\cos(\theta)\end{bmatrix}.$$

Proof for Theorem 15

Proof. By definition, the iterates of Algorithm 12 satisfy

$$\begin{aligned}\tfrac{1}{\gamma}(\mathbf{B}_k - \mathbf{B}_{k+1}) + \partial\psi(\mathbf{B}_k) - \partial\psi(\mathbf{B}_{k+1}) \ni &\mathbf{X}^\top(\mathbf{X}\mathbf{B}_k - \mathbf{X}\mathbf{A}_k) + \partial\psi(\mathbf{B}_k)\\ \mathbf{0} \in &(\mathbf{A}_{k+1}\mathbf{X}^\top\mathbf{B}_{k+1}^\top - \mathbf{X}^\top)\mathbf{B}_{k+1}\mathbf{X} + \partial\varphi(\mathbf{A}_{k+1}).\end{aligned}$$

From the definition of the objective, we have,

$$\begin{aligned}
f(\mathbf{A}_{k+1}, \mathbf{B}_{k+1}) =& \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}_{k+1}^\top\|_F^2 + \psi(\mathbf{B}_{k+1}) + \varphi(\mathbf{A}_{k+1}) \\
\leq& \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}_{k}^\top\|_F^2 + \varphi(\mathbf{A}_{k}) + \psi(\mathbf{B}_{k+1}) \\
=& \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k}\mathbf{A}_{k}^\top + \mathbf{X}\mathbf{B}_{k}\mathbf{A}_{k}^\top - \mathbf{X}\mathbf{B}_{k+1}\mathbf{A}_{k}^\top\|_F^2 + \varphi(\mathbf{A}_{k}) + \psi(\mathbf{B}_{k+1}) \\
=& \frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{B}_{k}\mathbf{A}_{k}^\top\|_F^2 + \langle \mathbf{X}(\mathbf{A}_k - \mathbf{B}_k)\rangle \, \mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1}) \\
&+ \frac{1}{2}\|\mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1})\|_F^2 + \psi(\mathbf{B}_{k+1}) + \varphi(\mathbf{A}_k) \\
=& f(\mathbf{A}_k, \mathbf{B}_k) + \langle \mathbf{X}(\mathbf{A}_k - \mathbf{B}_k)\rangle \, \mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1}) \\
&+ \frac{1}{2}\|\mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1})\|_F^2 + \psi(\mathbf{B}_{k+1}) - \psi(\mathbf{B}_k).
\end{aligned}$$

Since ψ is a convex function, we have,

$$\psi(\mathbf{B}_{k+1}) - \psi(\mathbf{B}_k) \leq \langle \partial\psi(\mathbf{B}_{k+1})\rangle \, \mathbf{B}_{k+1} - \mathbf{B}_k.$$

Therefore,

$$\begin{aligned}
f(\mathbf{A}_{k+1}, \mathbf{B}_{k+1}) - f(\mathbf{A}_k, \mathbf{B}_k) \leq& \left\langle \mathbf{X}^\top\mathbf{X}(\mathbf{A}_k - \mathbf{B}_k)\right\rangle \mathbf{B}_k - \mathbf{B}_{k+1} + \frac{1}{2}\|\mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1})\|_F^2 \\
&+ \left\langle \frac{1}{\gamma}(\mathbf{B}_k - \mathbf{B}_{k+1}) + \mathbf{X}^\top\mathbf{X}(\mathbf{A}_k - \mathbf{B}_k)\right\rangle \mathbf{B}_{k+1} - \mathbf{B}_k \\
=& -\frac{1}{\gamma}\|\mathbf{B}_k - \mathbf{B}_{k+1}\|_F^2 + \frac{1}{2}\|\mathbf{X}(\mathbf{B}_k - \mathbf{B}_{k+1})\|_F^2 \\
\leq& -\frac{1}{2}\|\mathbf{X}\|_2^2\|\mathbf{B}_k - \mathbf{B}_{k+1}\|_F^2.
\end{aligned}$$

Using the definition of optimality condition T, we have

$$\begin{aligned}
T(\mathbf{A}_k, \mathbf{B}_k) \leq& \left(\frac{1}{\|\mathbf{X}\|_2^2} + L\right)^2 \|\mathbf{B}_k - \mathbf{B}_{k+1}\|_F^2 \\
\leq& \frac{2(\|\mathbf{X}\|_2^2 + L)^2}{\|\mathbf{X}\|_2^2}(f(\mathbf{A}_k, \mathbf{B}_k) - f(\mathbf{A}_{k+1}, \mathbf{B}_{k+1}))
\end{aligned}$$

Adding up the terms across k, we have a telescoping series on the right hand side, and immediately obtain the result. □

Chapter 6

LEARNING ROBUST REPRESENTATIONS FOR COMPUTER VISION

Unsupervised learning techniques in computer vision often require learning latent representations, such as low-dimensional linear and non-linear subspaces. Noise and outliers in the data can frustrate these approaches by obscuring the latent spaces.

Our main goal is deeper understanding and new development of robust approaches for representation learning. We provide a new interpretation for existing robust approaches and present two specific contributions: a new robust PCA approach, which can separate foreground features from dynamic background, and a novel robust spectral clustering method, that can cluster facial images with high accuracy. Both contributions show superior performance to standard methods on real-world test sets.

6.1 Introduction

Supervised learning, and in particular deep learning [216, 307], have been very successful in computer vision. Applications include autoencoders [345] that map between noisy and clean images [361], convolutional networks for image/video analysis [204], and generative adversarial networks that synthesize real world-like images [169].

In contrast, unsupervised learning still poses significant challenges. Broadly, unsupervised learning seeks to discover hidden structure in the data without using ground truth labels, thereby revealing features of interest. In this paper, we consider unsupervised representation learning methods which can be used along with centroid-based clustering to summarize the data distribution using a few characteristic samples. We are interested in spectral clustering [260] and subspace clustering [134]; the proposed ideas can also be generalized to deep embedding-based clustering strategies [360]. *Spectral clustering* methods use neighborhood graphs to learn the underlying representation [260]; this approach is used for image segmentation [312, 375] and 3D mesh segmentation [227]. *Subspace clustering* methods model the dataset as a union of low-dimensional linear subspaces and utilize sparse and low-rank methods to obtain the representation; this model is used for facial clustering and recognition [134, 309].

Learning effective latent representations hinges on accurately modeling noise and outliers. Further, in practice, the data satisfy the structural assumptions (union of subspaces, low rank, etc.) only approximately. Adopting robust optimization strategies is a natural way to combat these challenges. For example, consider principal component analysis (PCA), a prototypical representation learning method based on matrix factorization. Given low-rank

data contaminated by outliers, the classical PCA method will fail to find it. Consequently, the robust PCA (rPCA) method [79], which decomposes data into low rank and sparse components, is preferred in practice, e.g. background/foreground separation [79, 315]. Similarly, when data assumed to be from a union of subspaces is contaminated by outliers, allowing for sparse outliers during optimization leads to accurate recovery of the subspaces, e.g. face classification [358].

Our goal is to develop effective robust formulations for unsupervised representation learning tasks in computer vision; we are interested in complex situations, when the data is corrupted with a combination of sparse outliers and dense noise.

Contributions. We first review the relationship between outlier models and statistically robust formulations. In particular, we show that the rPCA formulation is equivalent to solving a Huber regression problem for low-rank representation learning. Using this connection, we develop a new nonconvex penalty, dubbed the Tiber, designed to aggressively penalize mid-sized residuals. In Section 6.2, we show that this penalty is well suited for dynamic background separation, outperforming classic rPCA methods.

Our second contribution is to use the design philosophy behind robust low-rank representation learning to develop a new formulation for robust clustering. We formulate classic spectral analysis as an optimization problem, and then modify this problem to be robust to outliers. The advantages are shown using a synthetic clustering example. We then combine robust spectral clustering with robust subspace clustering to achieve superior performance on face recognition tasks, surpassing prior work without any data pre-processing; see Section 6.3, Table 6.1.

6.2 New Penalties for Learning Robust Representations

Many tasks in computer vision depend on unsupervised representation learning. A well-known example is background/foreground separation, often solved by robust principal component analysis (rPCA). rPCA learns low-rank representations by decomposing a data matrix into a sum of low-rank and sparse components. The low-rank component represents the background and the sparse component represents the foreground [79].

In this section, we show that rPCA is equivalent to a robust regression problem, and solving a Huber-robust regression [195] for the background representation is completely equivalent to the full rPCA solution. We use this equivalence to design a new robust penalty (dubbed Tiber) based on statistical descriptions of the signals of interest. We illustrate the benefits of using this new non-convex penalty for separating foreground from a dynamic background, using real datasets.

6.2.1 Huber in rPCA

Background/foreground separation is widely used for detecting moving objects in videos from stationary cameras. A broad range of techniques have been developed to tackle this

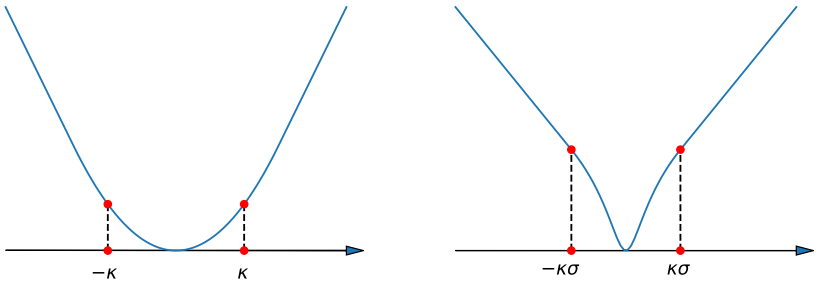


Figure 6.1: Robust penalties: left: Huber, right: Tiber. Both grow linearly outside an interval containing the origin. The Tiber penalizes 'mid-sized' errors within the region far more aggressively than the Huber; such a penalty must necessarily be non-convex.

task, ranging from simple thresholding [343] to mixtures of Gaussian models[318, 139, 177]. In particular, rPCA has been widely adopted to solve this problem [176, 270].

Denote a given video stream by $Y \in \mathcal{R}^{n\times m}$, where each of m frames is reshaped to be a vector of size n. There are many variants of rPCA [10]. We use the *stable principal component pursuit* (SPCP) formulation:

$$\min_{L,S} \frac{1}{2}\|L+S-Y\|_F^2 + \kappa\|S\|_1 + \lambda\|L\|_* \tag{6.1}$$

where L represents the background, and S the foreground. The regularizations used by this formulation ensure that L is chosen to be low rank, while S is designed to be sparse; using a quadratic penalty on the residual fits of the data up to some error level.

We can minimize over the variables in any order. Minimizing the first two summands of (6.1) in S gives a closed form function

$$\min_{S} \frac{1}{2}\|L+S-Y\|_F^2 + \kappa\|S\|_1 = \rho(L-Y;\kappa),$$

with $\rho(r;\kappa)$ the well-known Huber penalty [195]

$$\rho(r;\kappa) = \begin{cases} \kappa|r| - \kappa^2/2, & |r| > \kappa \\ r^2/2, & |r| \le \kappa \end{cases}. \tag{6.2}$$

We provide a simple statement of the following well-known result with a short self-contained proof.

Claim 1.

$$\rho(r;\kappa) = \min_{s} \frac{1}{2}(s-r)^2 + \kappa|s|. \tag{6.3}$$

Proof. The solution to this optimization problem is the *soft threshold* function (see e.g. [96])

$$\arg\min_s \frac{1}{2}(s-r)^2 + \kappa|s| = \mathbb{S}_\kappa(r) = \begin{cases} r-\kappa, & r > \kappa \\ 0, & |r| \le \kappa \\ r+\kappa, & r < -\kappa \end{cases}.$$

Plugging $\mathbb{S}_\kappa(r)$ back into (6.3), we have

$$\frac{1}{2}[\mathbb{S}_\kappa(r) - r]^2 + \kappa|\mathbb{S}_\kappa(r)| = \rho(r;\kappa).$$

□

The optimization problem is separable, so the result immediately extends to the vector case. Upon minimization over S, problem (6.1) then reduces to

$$\min_L \rho(L - Y;\kappa) + \lambda\|L\|_*. \tag{6.4}$$

To simplify the problem further, we use a factorized representation of L [64], choosing the rank to be $k \ll \min(n,m)$ to obtaining the non-convex formulation

$$\min_{U,V} \rho(U^\top V - Y;\kappa) \tag{6.5}$$

where $U \in \mathcal{R}^{k\times n}$ and $V \in \mathcal{R}^{k\times m}$.

Comparing (6.5) to (6.1) we see two advantages:

1. The dimension of the decision variable has been reduced from $2nm$ to $k(n+m)$.

2. (6.5) is smooth, and does not require computing SVDs.

Once we have U and V, we can easily recover L and S:

$$L = U^\top V, \quad S = \mathbb{S}_\kappa(U^\top V - Y).$$

The approach is illustrated in the left panels of Figure 6.2. Although the residual $U^T V - Y$ (shown in row 2) is noisy and not sparse, applying $\mathbb{S}_\kappa$ we get the sparse component (row 3), just as we would by solving the original formulation (6.1).

From a statistical perspective, the equivalence of rPCA and Huber means that the residual $R = U^\top V - Y$, which contains both S and random noise, can be modeled by a heavy tailed error distribution.

Claim 2. *Suppose* $\{r_i(x)\}_{i=1}^{l}$ *are i.i.d. samples from a distribution with density*

$$p(r;\theta) = \frac{1}{n_c(\theta)} \exp[-\rho(r;\theta)]$$

where $n_c(\theta) = \int_{\mathcal{R}} \exp[-\rho(r;\theta)]\, dr$ *is the normalization constant. Then maximum likelihood formulation for* x *is equivalent to the minimization problem*

$$\min_x \sum_{i=1}^{l} \rho(r_i(x);\theta).$$

The claim follows immediately by taking the negative log of the maximum likelihood. Claim 2 means that solving (6.5) is equivalent to assuming that elements in $R = U^\top V - Y$ are i.i.d. samples from the Laplace density

$$p(r;\kappa) = \frac{1}{n_c(\kappa)} \exp[-\rho(r;\kappa)].$$

The function ρ has linear tails (See Figure 6.1), which means this distribution is much more likely to produce large samples compared to the Gaussian.

6.2.2 Weaknesses of the Huber

Although the Huber distribution can detect sparse outliers, it does not model small errors well. In many background/foreground separation problems, we must cope with a dynamic background (e.g. motion of tree leaves or water waves). These small dynamic background perturbations correspond to motion we do not care about — we are much more interested in detecting cars, people, and animals moving through the scene.

We want to move these dynamics into the low-rank background term. However, the Huber is quadratic near the origin (i.e. nearly flat), so small perturbations do not significantly affect the objective value; and solving (6.5) leaves these terms in the residual R. Thresholding these terms is either too aggressive (removing features we care about), or too lenient, leaving the dynamics in the foreground (see first two columns of Figure 6.2). A better penalty would rise steeply for small values of R, without significantly affecting tail behavior.

6.2.3 Tiber for rPCA

We propose a new penalty, which we call the Tiber. While the Huber is defined by partially minimizing the sum of the 1-norm with a quadratic (6.2), the Tiber replaces the quadratic with a nonconvex function. The resulting penalty can match the tail behavior of Huber, yet have different properties around the origin (see Figure 6.1). Tiber is better suited for background/foreground separation problems with dynamic background. We define the penalty

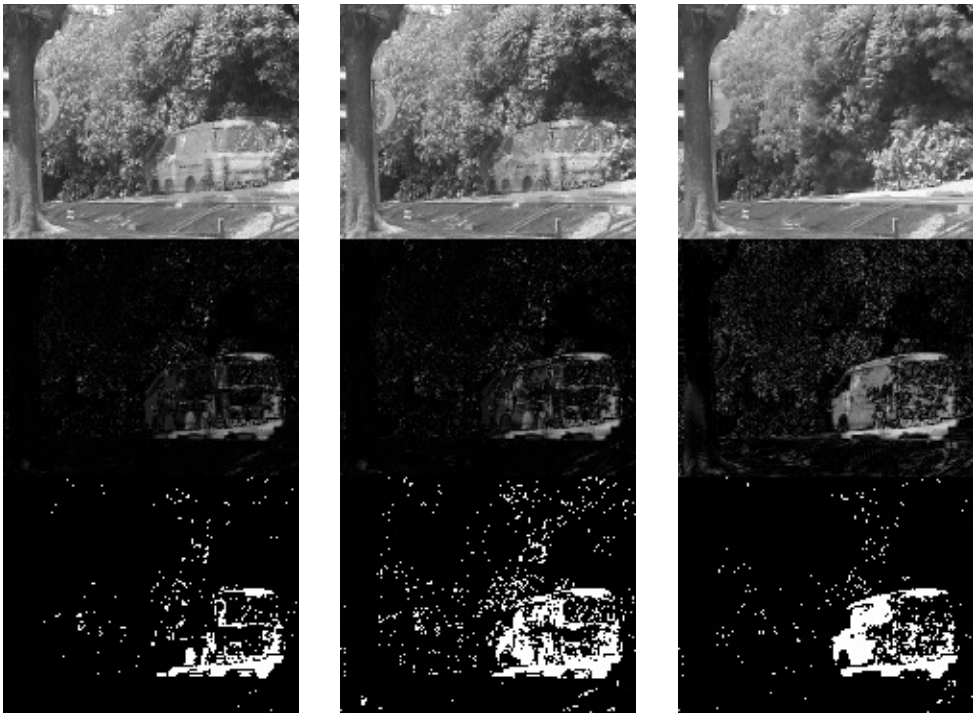


Figure 6.2: Left: Huber with $\kappa = 0.15$, middle: Huber with $\kappa = 0.1$, right: Tiber with $\kappa = 10, \sigma = 0.03$. Row 1: low rank component L, row 2: residual $|R| = |U^\top V - Y|$, row 3: binary plot for S. The Tiber recovers the van while avoiding the dynamic background.

as follows:

$$\rho_{\mathrm{T}}(r; [\kappa, \sigma]) = \begin{cases} \frac{2\kappa}{\sigma(\kappa^2+1)}(|r| - \kappa\sigma) + \log(1+\kappa^2), & |r| > \kappa\sigma \\ \log(1 + r^2/\sigma^2), & |r| \leq \kappa\sigma \end{cases} \tag{6.6}$$

The Tiber is parametrized by thresholding parameter κ and scale parameter σ. Just as the Huber, it can be expressed as the value function of a minimization problem. We replace the quadratic penalty in Claim 1 by the smooth nonconvex penalty $\log(1 + (\cdot)^2)$. For simplicity, we use $\sigma = 1$ in the result below.

Claim 3.

$$\rho_{\mathrm{T}}(r; [\kappa, 1]) = \min_s \log(1 + (s-r)^2) + \frac{2\kappa}{1+\kappa^2}|s|. \tag{6.7}$$

Proof. Denote the objective function in (6.7) by $f(s)$. It is easy to check that f is quasi-

convex in s when $\kappa \geq 0$. We look to local optimality conditions to understand the structure of the minimizers.

- Suppose $s^* > 0$. Then $0 = f'(s^*)$ means

$$0 = \frac{2(s^* - r)}{1 + (s^* - r)^2} + \frac{2\kappa}{1 + \kappa^2} \iff s^* = r - \kappa;$$

  this requires $r > \kappa$.

- Suppose $s^* < 0$. Then $0 = f'(s^*)$ means

$$0 = \frac{2(s^* - r)}{1 + (s^* - r)^2} + \frac{-2\kappa}{1 + \kappa^2} \iff s^* = r + \kappa;$$

  this requires $r < -\kappa$.

- otherwise $s^* = 0$.

Therefore $s^* = \mathbb{S}_\kappa(r)$. Plugging this into (6.7), we have

$$\rho_{\mathrm{T}}(r; [\kappa, 1]) = \log(1 + (\mathbb{S}_\kappa(r) - r)^2) + \frac{2\kappa}{1 + \kappa^2}|\mathbb{S}_\kappa(r)|.$$

□

In Figure 6.1, we see that Tiber rises steeply near the origin. This behavior discourages dynamic terms (leaves, waves) in R, forcing them to be fit by $U^\top V$. The new Tiber-robust rPCA problem is given by:

$$\min_{U,V} \rho_{\mathrm{T}}(U^\top V - Y; [\kappa, \sigma]) \tag{6.8}$$

which also has all of the advantages of (6.5). Moreover, because of the characterization from Claim 3, once we solve (6.8), we immediately recover L and S:

$$L = U^\top V, \quad S = \mathbb{S}_{\kappa\sigma}(U^\top V - Y).$$

6.2.4 Experiment: Foreground Separation

We use a publicly available data set[1] with a dynamic background (moving trees). We sample 102 data frames from this data set, convert them to grey scale, and reshape them as column vectors of matrix $Y \in \mathcal{R}^{20480 \times 102}$. We compare formulations (6.5) and (6.8). Proximal alternating linearized minimization algorithm (PALM) [52] was used to solve all of the optimization problems.

[1]Downloaded from `http://vis-www.cs.umass.edu/~narayana/castanza/I2Rdataset/`

Rank of U and V was set to be $k = 10$ for all experiments. We manually tuned parameters to achieve the best possible recovery in each formulation. For Huber, we selected two nearby κ values, $\kappa = 0.15$ and $\kappa = 0.1$; for Tiber, we selected $\kappa = 10$ and $\sigma = 0.03$, resulting in the threshold parameter $\kappa\sigma = 0.3$.

The results are shown in Figure 6.2. The task is identifying the van while avoiding interference from moving leaves. The Huber is unable to separate the van from the leaves for any threshold values κ. When we choose $\kappa = 0.15$ (left panel in Figure 6.2), we cut out too much information, giving an incomplete van in S. If we make a less conservative choice $\kappa = 0.1$ (middle panel in Figure 6.2), we leave too much dynamic noise in S, which obscures the van.

The Tiber Penalty obtains a cleaner picture of the moving vehicle (right panel in Figure 6.2). As expected, it forces more of the dynamic background to be fit by $U^\top V$, leaving a fairly complete van in S without too much contamination.

6.3 Robust Representation Learning for Clustering

Centroid-based clustering, e.g. k-Means, is a standard tool to partition and summarize datasets. Given the high dimensionality and complexity of data in computer vision applications, it is necessary to learn latent representations, such as the underlying metric, prior to clustering. Clustering is then performed in the latent space.

We develop an approach for robust spectral clustering. We illustrate the advantages using a synthetic dataset, and then combine the approach with robust subspace clustering to achieve perfect performance on face recognition tasks.

6.3.1 Spectral Clustering

Spectral clustering [260] is formulated as follows. Given m datapoints $y_i \in \mathbb{R}^n$, we arrange them in a matrix $Y \in \mathcal{R}^{n\times m}$. To partition the data into k groups, spectral clustering uses the following steps:

1. Given a dataset of m samples, we construct the similarity matrix $L \in \mathcal{R}^{m\times m}$ of the data points.

2. Extract the eigenvectors $X \in \mathcal{R}^{m\times k}$ of L corresponding to the k largest eigenvalues.

3. Project each row of X onto the unit ball, and apply distance-based clustering.

Finding a meaningful similarity matrix L is crucial to the success of spectral clustering. Ideally, L will be a block diagonal matrix with k blocks. This rarely happens for real applications; even when underlying structure in L is present, it can be obscured by noise and a small number of points that don't follow the general pattern.

To find a factorization of noisy L, we need a robust method for eigenvalue decomposition. We first formulate eigenvalue decomposition as an optimization problem.

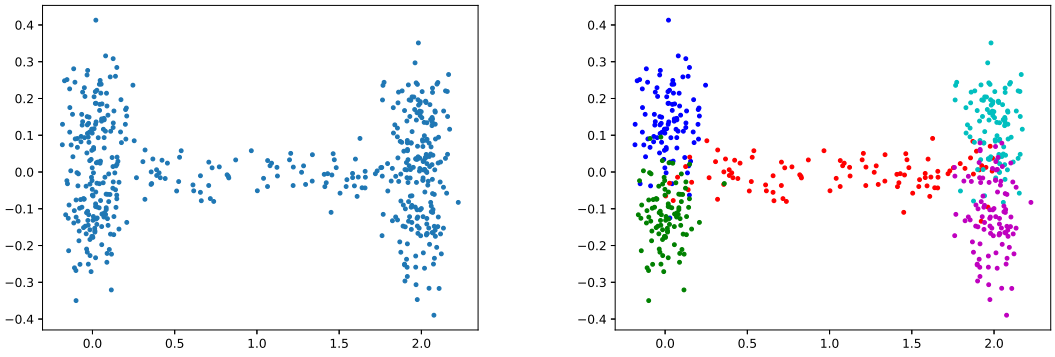


Figure 6.3: Synthetic Data Clustering: Up: data without labels, Down: data with true colors.

Claim 4. *Assume L is a symmetric matrix with eigenvalues less than or equal to 1. Then the solution to the problem*

$$\begin{aligned} \min_{X} \ & \frac{1}{2}\|XX^\top - L\|_F^2 \\ s.t. \ & X^\top X = I_k \end{aligned} \tag{6.9}$$

is $X = [v_1, \ldots, v_k]$ with v_i the eigenvector corresponding to the i^{th} largest eigenvalue of L, and I_k the k by k identity matrix.

Proof. Since L is a symmetric matrix, it has a eigenvalue decomposition,

$$L = Y\Lambda Y^\top$$

where $Y \in \mathcal{R}^{m\times m}$ is orthogonal and Λ is diagonal, with $1 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. Similarly, we have

$$XX^\top = \tilde{X}D\tilde{X}^\top$$

where $\tilde{X} \in \mathcal{R}^{m\times m}$ is a orthogonal matrix whose first k columns agree with those of X, D is a diagonal matrix with first k elements on the diagonal are 1 and the rest are 0. From the Cauchy-Schwarz inequality, we have

$$\text{trace}(XX^\top \cdot L) = \left\langle XX^\top, L \right\rangle \leq \|XX^\top\|_F \cdot \|L\|_F$$

where equality hold when $XX^\top$ and L share the same singular vectors, i.e., X equals to the first k columns of Y.

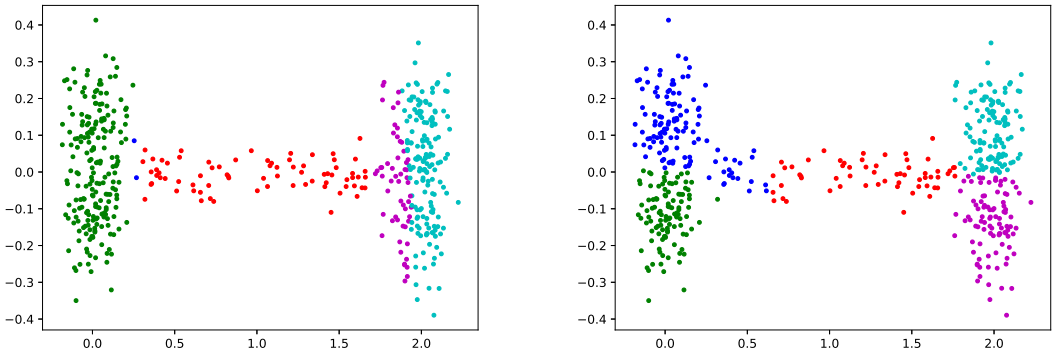


Figure 6.4: Synthetic Data Clustering: Up: result from eigenvalue decomposition, Down: result from (6.10).

Therefore

$$\begin{aligned}\frac{1}{2}\|XX^\top - L\|_F^2 &= \frac{1}{2}\|XX^\top\|_F^2 - \left\langle XX^\top, L\right\rangle + \frac{1}{2}\|L\|_F^2 \\ &\geq \frac{1}{2}\|D\|_F^2 - \|D\|_F\|\Lambda\|_F + \frac{1}{2}\|\Lambda\|_F^2\end{aligned}$$

with equality hold when columns of X are eigenvectors corresponding to the largest k eigenvalues. □

We robustify (6.9) by replacing the Frobenius norm in the optimization formulation by the Huber function (or another robust penalty):

$$\begin{aligned}\min_X \ & \rho(XX^\top - L; \kappa) \\ \text{s.t. } & X^\top X = I_k\end{aligned}. \tag{6.10}$$

This approach can be very effective. Consider the following clustering experiment with $n = 2$, $m = 500$, and $k = 5$. We generate five clusters (sampling from four 2-D Gaussians, one rectangular uniform distribution) with 100 points per group. To make the problem challenging, we move the clusters close together so much that trying to tell them apart with the naked eye is hard (Figure 6.3, top). True clusters appear in Figure 6.3, bottom.

Classic spectral clustering, which uses eigenvalue decomposition in step 2, fails to detect the true relationships (Figure 6.4, top). Robust spectral clustering using the Huber penalty (6.10) does a much better job (Figure 6.4, bottom).

6.3.2 Subspace Clustering

Subspace clustering looks for low dimensional representation of high dimensional data, by grouping the points along low-dimensional subspaces. Given a data matrix $Y \in \mathcal{R}^{n \times m}$ as in Section 6.3.1, the optimization for subspace clustering is given by [134]:

$$\min_{C} \ \frac{1}{2}\|Y - YC\|_F^2 + \lambda\|C\|_1 \ \text{ s.t. } \ \text{diag}(C) = 0. \tag{6.11}$$

This formulation looks for a sparse representation of the dataset by its members: $s_i = Sc_i$. To avoid the trivial solution, we require the diagonal of C to be identically 0. After obtaining C, it is post-processed and a similarity matrix is constructed as $W = |C| + |C^\top|$. W will be ideally close to block-diagonal, where each block represents a subspace, and spectral clustering is performed it to identify cluster memberships.

Outliers in the dataset can break the performance of (6.11). To make the approach robust, [134] uses the formulation

$$\begin{aligned} \min_{C} \ & \frac{1}{2}\|Y - YC - S\|_F^2 + \lambda\|C\|_1 + \kappa\|S\|_1 \\ \text{s.t. } & \text{diag}(C) = 0. \end{aligned} \tag{6.12}$$

Using Claim 1, we rewrite (6.12) using Huber:

$$\min_{C} \ \rho(Y - YC; \kappa) + \lambda\|C\|_1 \ \text{ s.t. } \ \text{diag}(C) = 0. \tag{6.13}$$

Formulation (6.13) has the same advantages with respect to (6.12) as (6.5) has with respect to (6.1).

6.3.3 Face Clustering

Given multiple face images taken at different conditions, the goal of face clustering [134] is to identify images that belong to the same person.

We use images from the publicly available Extended Yale B dataset [219] [2]. Each image has 32×32 pixels, and there are 2414 images in the dataset. These images belong to 38 people, with approximately 64 pictures per person.

Under the Lambertian assumption, pictures obtained from one person under different illuminations should lie close to a 9 dimensional subspace [31]. In practice, these spaces are hard to detect because of noise in the images, and a robust approach is required.

Robust subspace clustering for face images:

1. Obtain sparse representation C using (6.13).

2. Construct similarity matrix W from C.

[2] Downloaded from `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`

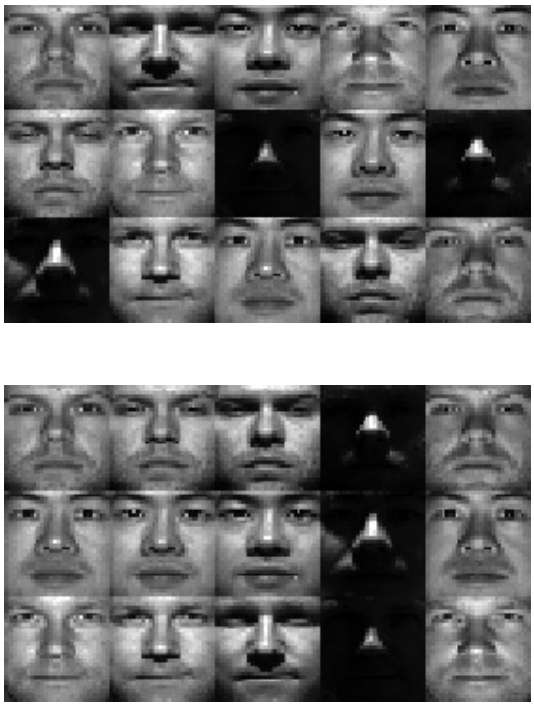


Figure 6.5: Faces data: top: randomly chosen face images, bottom: faces after clustering; each row belongs to a cluster.

- Normalize columns of C to have maximum absolute value no larger than 1.
- Form $W = |C| + |C^\top|$
- Normalize W: $W \leftarrow D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

3. Apply spectral clustering using W.
   - Apply robust symmetric factorization (6.10) to W, to obtain the latent representation X.
   - Project each row of X onto the unit 2-norm ball.
   - Apply K-means algorithm to the new rows of X.

The results are shown in Table 6.1. We implement the approach for different numbers of subjects $k = 2, 3, 5, 8$. We show the parameters κ and λ in (6.13) used to achieve the high

accuracies given in Table 6.1[3].

clusters	κ in (6.13)	λ in (6.13)	error	error in [134]
$k = 2$	0.5	1	0.00%	1.86%
$k = 3$	0.1	0.7	0.00%	3.10%
$k = 5$	0.05	0.7	0.00%	4.31%
$k = 8$	0.03	0.5	2.73%	5.85%

Table 6.1: Results for robust subspace clustering with face images.

To get better intuition of the method, we plot the similarity matrix corresponding to $k = 3$ in Figure 6.6. We can clearly see three blocks along the diagonal that correspond to the three face clusters. The resulting projected X obtained from the eigenvalue decomposition

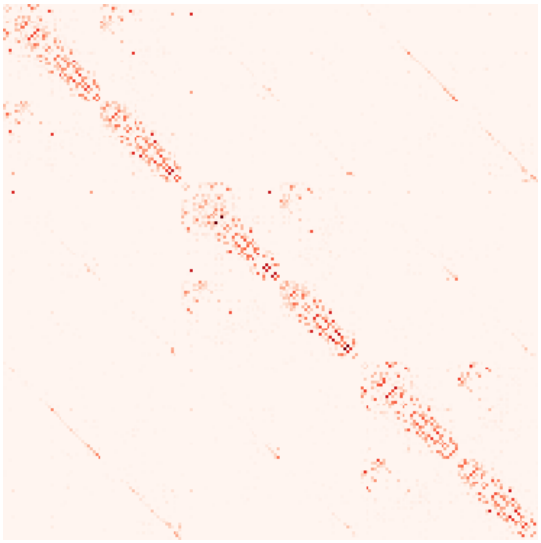


Figure 6.6: Similarity matrix for face images clustering with $k = 3$; the matrix is nearly block diagonal with 3 blocks.

of similarity matrix W are shown in Figure 6.7. The three clusters are clearly well separated. The final algorithm has perfect accuracy in this example.

[3]In [134], the images used are of size 48×42. The numbers shown are therefore indicative.

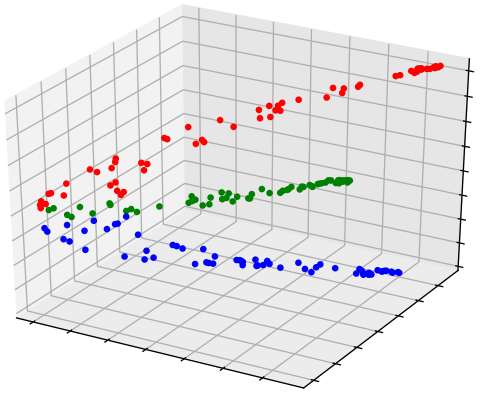


Figure 6.7: Projections of the rows of X onto the eigenspace of the similarity matrix for $k = 3$. Each color represent the face images of a single person.

6.4 Discussion

Robust approaches are essential for unsupervised learning, and can be designed using optimization formulations. For example, in both rPCA and robust spectral learning, SVD and eigenvalue decomposition are first characterized using optimization, then reformulated with robust losses.

Several tasks in this approach are difficult. First, there is a need to tune parameters in the optimization formulations. For example, the Tiber depends on two parameters, κ and σ. Automatic ways to tune these parameters can make robust unsupervised learning a lot more portable. Second, the optimization problems we have to solve are large-scale; time required for robust subspace clustering for images scales non-linearly with both the number and size of images. Designing non-smooth stochastic algorithms that take the structure of these problems into account is essential.

Chapter 7

TRIMMING THE ℓ_1 REGULARIZER

We study high-dimensional estimators with the trimmed ℓ_1 penalty, which leaves the h largest parameter entries penalty-free. While optimization techniques for this nonconvex penalty have been studied, the statistical properties have not yet been analyzed. We present the first statistical analyses for M-estimation, and characterize support recovery, ℓ_∞ and ℓ_2 error of the trimmed ℓ_1 estimates as a function of the trimming parameter h. Our results show different regimes based on how h compares to the true support size. Our second contribution is a new algorithm for the trimmed regularization problem, which has the same theoretical convergence rate as difference of convex (DC) algorithms, but in practice is faster and finds lower objective values. Empirical evaluation of ℓ_1 trimming for sparse linear regression and graphical model estimation indicate that trimmed ℓ_1 can outperform vanilla ℓ_1 and non-convex alternatives. Our last contribution is to show that the trimmed penalty is beneficial beyond M-estimation, and yields promising results for two deep learning tasks: input structures recovery and network sparsification.

7.1 Introduction

We consider high-dimensional estimation problems, where the number of variables p can be much larger that the number of observations n. In this regime, consistent estimation can be achieved by imposing low-dimensional structural constraints on the estimation parameters. *Sparsity* is a prototypical structural constraint, where at most a small set of parameters can be non-zero. A key class of sparsity-constrained estimators is based on regularized M-estimators using *convex* penalties, with the ℓ_1 penalty by far the most common. In the context of linear regression, the Lasso estimator [328] solves an ℓ_1 regularized or constrained least squares problem, and has strong statistical guarantees, including prediction error consistency [338], consistency of the parameter estimates in some norm [338, 250, 76], and variable selection consistency [249, 348, 382]. In the context of sparse Gaussian graphical model (GMRF) estimation, the graphical Lasso estimator minimizes the Gaussian negative log-likelihood regularized by the ℓ_1 norm of the off-diagonal entries of the concentration [370, 152, 29]. Strong statistical guarantees for this estimator have been established (see [287] and references therein).

Recently, there has been significant interest in *non-convex* penalties to alleviate the bias incurred by convex approaches, including SCAD and MCP penalties [140, 56, 376, 377]. In particular, [377] established consistency for the global optima of least-squares problems with certain non-convex penalties. [230] showed that under some regularity conditions on the

penalty, any stationary point of the objective function will lie within statistical precision of the underlying parameter vector and thus provide ℓ_2- and ℓ_1- error bounds for any stationary point. [231] proved that for a class of *amenable* non-convex regularizers with vanishing derivative away from the origin (including SCAD and MCP), any stationary point is able to recover the parameter support without requiring the typical incoherence conditions needed for convex penalties. All of these analyses apply to non-convex penalties that are *coordinate-wise separable*.

Our starting point is a family of M-estimators with trimmed ℓ_1 regularization, which leaves the largest h parameters unpenalized. This non-convex family includes the Trimmed Lasso [170, 43] as a special case. Unlike SCAD and MCP, trimmed regularization exactly solves constrained best subset selection for large enough values of the regularization parameter, and offers more direct control of sparsity via the parameter h. While Trimmed Lasso has been studied from an optimization perspective and with respect to its connections to existing penalties, it has *not* been analyzed from a statistical standpoint.

Contributions:

- We present the *first* statistical analysis of M-estimators with trimmed regularization, *including* Trimmed Lasso. Existing results for non-convex regularizers [230, 231] cannot be applied as trimmed regularization is neither coordinate-wise decomposable nor "ameanable". We provide support recovery guarantees, ℓ_∞ and ℓ_2 estimation error bounds for general M-estimators, and derive specialized corollaries for linear regression and graphical model estimation. Our results show different regimes based on how the trimming parameter h compares to the true support size.

- To optimize the trimmed regularized problem we develop and analyze a new algorithm, which performs better than difference of convex (DC) functions optimization [205].

- Experiments on sparse linear regression and graphical model estimation show ℓ_1 trimming is competitive with other non-convex penalties and vanilla ℓ_1 when h is selected by cross-validation, and has consistent benefits for a wide range of values for h.

- Moving beyond M-estimation, we apply trimmed regularization to two deep learning tasks: (i) recovering input structures of deep models and (ii) network pruning (a.k.a. sparsification, compression). Our experiments on input structure recovery are motivated by [271], who quantify complexity of sparsity encouraging regularizers by introducing the covering dimension, and demonstrates the benefits of regularization for learning over-parameterized networks. We show trimmed regularization achieves superior sparsity pattern recovery compared to competing approaches. For network pruning, we illustrate the benefits of trimmed ℓ_1 over vanilla ℓ_1 on MNIST classification using the LeNet-300-100 architecture. Next, motivated by recently developed pruning methods based on variational Bayesian approaches [104, 233], we propose Bayesian neural networks with trimmed ℓ_1 regularization. In our experiments, these achieve

superior results compared to competing approaches with respect to both error and sparsity level. Our work therefore indicates broad relevance of trimmed regularization in multiple problem classes.

7.2 Trimmed Regularization

Trimming has been typically applied to the *loss* function $\mathcal{L}$ of M-estimators. We can handle outliers by trimming *observations* with large residuals in terms of $\mathcal{L}$: given a collection of n samples, $\mathcal{D} = \{Z_1, \ldots, Z_n\}$, we solve

$$\underset{\boldsymbol{\theta}\in\Omega, \boldsymbol{w}\in\{0,1\}^n}{\text{minimize}} \sum_{i=1}^{n} w_i \mathcal{L}(\boldsymbol{\theta}; Z_i) \quad \text{s.t.} \sum_{i=1}^{n} w_i = n - h,$$

where Ω denotes the parameter space (e.g., $\mathbb{R}^p$ for linear regression). This amounts to trimming h outliers as we learn $\boldsymbol{\theta}$ (see [366] and references therein).

In contrast, we consider here a family of M-estimators with trimmed *regularization* for general high-dimensional problems. We trim entries of $\boldsymbol{\theta}$ that incur the largest penalty using the following program:

$$\begin{aligned} \underset{\boldsymbol{\theta}\in\Omega,\, \boldsymbol{w}\in[0,1]^p}{\text{minimize}} \quad & \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_n \sum_{j=1}^{p} w_j |\theta_j| \\ \text{s.t.} \quad & \mathbf{1}^\top \boldsymbol{w} \geq p - h\,. \end{aligned} \tag{7.1}$$

Defining the order statistics of the parameter $|\theta_{(1)}| > |\theta_{(2)}| > \ldots > |\theta_{(p)}|$, we can partially minimize over $\boldsymbol{w}$ (setting w_i to 0 or 1 based on the size of $|\theta_i|$), and rewrite the reduced version of problem (7.1) in $\boldsymbol{\theta}$ alone:

$$\underset{\boldsymbol{\theta}\in\Omega}{\text{minimize}} \quad \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_n \mathcal{R}(\boldsymbol{\theta}; h) \tag{7.2}$$

where the regularizer $\mathcal{R}(\boldsymbol{\theta}; h)$ is the smallest $p-h$ absolute sum of $\boldsymbol{\theta}$: $\sum_{j=h+1}^{p} |\theta_{(j)}|$. The constrained version of (7.2) is equivalent to minimizing a loss subject to a sparsity penalty [170]: $\text{minimize}_{\boldsymbol{\theta}\in\Omega}\, \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$ s. t. $\|\boldsymbol{\theta}\|_0 \leq h$. For statistical analysis, we focus on the reduced problem (7.2). When optimizing, we exploit the structure of (7.1), treating weights $\boldsymbol{w}$ as auxiliary optimization variables, and derive a new fast algorithm with a custom analysis that does not use DC structure. We focus on two key examples: sparse linear models and sparse graphical models. We also present empirical results for trimmed regularization of deep learning tasks to show that the ideas and methods generalize well to these areas.

Example 1: Sparse linear models. In high-dimensional linear regression, we observe n pairs of a real-valued target $y_i \in \mathbb{R}$ and its covariates $\boldsymbol{x}_i \in \mathbb{R}^p$ in a linear relationship:

$$\boldsymbol{y} = X\boldsymbol{\theta}^* + \boldsymbol{\epsilon}. \tag{7.3}$$

Here, $\boldsymbol{y} \in \mathbb{R}^n$, $X \in \mathbb{R}^{n\times p}$ and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a vector of n independent observation errors. The goal is to estimate the k-sparse vector $\boldsymbol{\theta}^* \in \mathbb{R}^p$. According to (7.2), we use the least squares loss function with trimmed ℓ_1 regularizer (instead of the standard ℓ_1 norm in Lasso [328]):

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^p}{\text{minimize}}\, \frac{1}{n}\big\|X\boldsymbol{\theta} - \boldsymbol{y}\big\|_2^2 + \lambda_n \mathcal{R}(\boldsymbol{\theta}; h). \tag{7.4}$$

Example 2: Sparse graphical models. GGMs form a powerful class of statistical models for representing distributions over a set of variables [215], using undirected graphs to encode conditional independence conditions among variables. In the high-dimensional setting, graph sparsity constraints are particularly pertinent for estimating GGMs. The most widely used estimator, the graphical Lasso minimizes the negative Gaussian log-likelihood regularized by the ℓ_1 norm of the entries (or the off-diagonal entries) of the precision matrix (see [370, 152, 29]). In our framework, we replace ℓ_1 norm with its trimmed version: $\text{minimize}_{\boldsymbol{\Theta}\in\mathcal{S}_{++}^p}\ \text{trace}\big(\widehat{\Sigma}\boldsymbol{\Theta}\big) - \log\det\big(\boldsymbol{\Theta}\big) + \lambda_n\mathcal{R}(\boldsymbol{\Theta}_{\text{off}}; h)$ where $\mathcal{S}_{++}^p$ denotes the convex cone of symmetric and strictly positive definite matrices, $\mathcal{R}(\boldsymbol{\Theta}_{\text{off}}; h)$ does the smallest $p(p-1)-h$ absolute sum of off-diagonals.

Relationship with SLOPE (OWL) penalty. Trimmed regularization has an apparent resemblance to the SLOPE (or OWL) penalty [51, 148], but the two are in fact distinct and pursue different goals. Indeed, the SLOPE penalty can be written as $\sum_{i=1}^p w_i|\beta_{(i)}|$ for a *fixed* set of weights $w_1 \geq w_2 \geq \cdots \geq w_p \geq 0$ and where $|\beta_{(1)}| > |\beta_{(2)}| > \cdots > |\beta_{(p)}|$ are the sorted entries of $\boldsymbol{\beta}$. SLOPE is convex and penalizes more those parameter entries with *largest amplitude*, while trimmed regularization is generally non-convex, and only penalizes entries with *smallest amplitude*; the weights are also optimization variables. While the goal of trimmed regularization is to alleviate bias, SLOPE is akin to a significance test where top ranked entries are subjected to a "tougher" threshold, and has been employed for clustering strongly correlated variables [148]. Finally from a robust optimization standpoint, Trimmed regularization can be viewed as using an optimistic (min-min) model of uncertainty and SLOPE a pessimistic (min-max) counterpart. We refer the interested reader to [43] for an in-depth exploration of these connections.

Relationship with ℓ_0 regularization. The ℓ_0 norm can be written as $\|\boldsymbol{\theta}\|_0 = \sum_{j=1}^p z_j$ with reparameterization $\theta_j = z_j\tilde{\theta}_j$ such that $z_j \in \{0,1\}$ and $\tilde{\theta}_j \neq 0$. [233] suggest a smoothed version via continuous relaxation on $\boldsymbol{z}$ in a variational inference framework. The variable $\boldsymbol{z}$ plays a similar role to $\boldsymbol{w}$ in our formulation in that they both learn sparsity patterns. In Section 7.4 we consider a Bayesian extension of the trimmed regularization problem where $\boldsymbol{\theta}$ only is be treated as Bayesian, since we can optimize $\boldsymbol{w}$ without any approximation, in contrast to previous work which needs to relax the discrete nature of $\boldsymbol{z}$.

7.3 Statistical Guarantees of M-Estimators with Trimmed Regularization

Our goal is to estimate the *true* k-sparse parameter vector (or matrix) $\boldsymbol{\theta}^*$ that is the minimizer of expected loss: $\boldsymbol{\theta}^* := \mathrm{argmin}_{\boldsymbol{\theta}\in\Omega} \mathbb{E}[\mathcal{L}(\boldsymbol{\theta})]$. We use S to denote the support set of $\boldsymbol{\theta}^*$, namely the set of non-zero entries (i.e., $k = |S|$). In this section, we derive support recovery, ℓ_∞ and ℓ_2 guarantees under the following standard assumptions:

(C-1) The loss function $\mathcal{L}$ is differentiable and convex.

(C-2) (Restricted strong convexity on $\boldsymbol{\theta}$) Let $\mathbb{D}$ be the possible set of error vector on the parameter $\boldsymbol{\theta}$. Then, for all $\Delta := \boldsymbol{\theta} - \boldsymbol{\theta}^* \in \mathbb{D}$, $\left\langle \nabla\mathcal{L}(\boldsymbol{\theta}^* + \Delta) - \nabla\mathcal{L}(\boldsymbol{\theta}^*), \Delta \right\rangle \geq \kappa_l \|\Delta\|_2^2 - \tau_1 \frac{\log p}{n} \|\Delta\|_1^2$, where κ_l is a "curvature" parameter, and τ_1 is a "tolerance" constant.

In the high-dimensional setting ($p > n$), the loss function $\mathcal{L}$ cannot be strongly convex in general. (C-2) imposes strong curvature only in some limited directions where the ratio $\frac{\|\Delta\|_1}{\|\Delta\|_2}$ is small. This condition has been extensively studied and known to hold for several popular high dimensional problems (see [286, 257, 230] for instance). The convexity condition of $\mathcal{L}$ in (C-1) can be relaxed as shown in [231]. For clarity, however, we focus on convex loss functions.

We begin with ℓ_∞ guarantees. We use a primal-dual witness (PDW) proof technique, which we adapt to the trimmed regularizer $\mathcal{R}(\boldsymbol{\theta}; h)$. The PDW method has been used to analyze the support set recovery of ℓ_1 regularization [349, 365] as well as decomposable and amenable non-convex regularizers [231]. However, the trimmed regularizer $\mathcal{R}(\boldsymbol{\theta}; h)$ is neither decomposable nor amenable, thus the results of [231] cannot be applied. The key step of PDW is to build a restricted program: Let T be an arbitrary subset of $\{1, \ldots, p\}$ of size h. Denoting $U := S \cup T$ and $V := S - T$, we consider the following restricted program: $\widehat{\boldsymbol{\theta}} \in \mathrm{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^U:\ \boldsymbol{\theta}\in\Omega}\ \mathcal{L}(\boldsymbol{\theta}) + \lambda_n \mathcal{R}(\boldsymbol{\theta}; h)$ where we fix $\widehat{\boldsymbol{\theta}}_j = 0$ for all $j \in U^c$. We further construct the dual variable $\widehat{\boldsymbol{z}}$ to satisfy the zero sub-gradient condition

$$\nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}) + \lambda_n \widehat{\boldsymbol{z}} = 0 \tag{7.5}$$

where $\widehat{\boldsymbol{z}} = (0, \widehat{\boldsymbol{z}}_V, \widehat{\boldsymbol{z}}_{U^c})$ for $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\theta}}_T, \widehat{\boldsymbol{\theta}}_V, 0_{U^c})$ (after re-ordering indices properly) and $\widehat{\boldsymbol{z}}_V \in \partial\|\widehat{\boldsymbol{\theta}}_V\|_1$. We suppress the dependency on T in $\widehat{\boldsymbol{z}}$ and $\widehat{\boldsymbol{\theta}}$ for clarity. In order to derive the final statement, we will establish the strict dual feasibility of $\widehat{\boldsymbol{z}}_{U^c}$, i.e., $\|\widehat{\boldsymbol{z}}_{U^c}\|_\infty < 1$.

The following theorem describes our main theoretical result concerning *any* local optimum of the non-convex program (7.2). The theorem guarantees under strict dual feasibility that non-relevant parameters of local optimum have smaller absolute values than relevant parameters; hence relevant parameters are not penalized (as long as $h \geq k$).

Theorem 17. *Consider the problem with trimmed regularizer* (7.2) *that satisfies* (C-1) *and* (C-2). *Let* $\widetilde{\boldsymbol{\theta}}$ *be an any local minimum of* (7.2) *with a sample size* $n \geq \frac{2\tau_1}{\kappa_l}(k+h)\log p$ *and* $\lambda_n \geq 2\|\nabla\mathcal{L}(\boldsymbol{\theta}^*)\|_\infty$. *Suppose that:*

(a) given **any** *selection of* $T \subseteq \{1, \ldots, p\}$ *s.t.* $|T| = h$*, the dual vector* $\widehat{\boldsymbol{z}}$ *from the PDW construction* (7.5) *satisfies the strict dual feasibility with some* $\delta \in (0, 1]$*,* $\|\widehat{\boldsymbol{z}}_{U^c}\|_\infty \leq 1 - \delta$ *where* U *is the union of true support* S *and* T*,*

(b) letting $\widehat{Q} := \int_0^1 \nabla^2 \mathcal{L}\big(\boldsymbol{\theta}^* + t(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\big)dt$*, the minimum absolute value* $\boldsymbol{\theta}^*_{\min} := \min_{j \in S} |\boldsymbol{\theta}^*_j|$ *is lower bounded by*

$\frac{1}{2}\boldsymbol{\theta}^*_{\min} \geq \|(\widehat{Q}_{UU})^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U\|_\infty + \lambda_n |\!|\!|(\widehat{Q}_{UU})^{-1}|\!|\!|_\infty$ *where* $|\!|\!|\cdot|\!|\!|_\infty$ *denotes the maximum absolute row sum of the matrix.*

Then, the following properties hold:

(1) For every pair $j_1 \in S, j_2 \in S^c$*, we have* $|\widetilde{\boldsymbol{\theta}}_{j_1}| > |\widetilde{\boldsymbol{\theta}}_{j_2}|$*,*

(2) If $h < k$*, all* $j \in S^c$ *are successfully estimated as zero and* $\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty$ *is upper bounded by*

$$\big\|\big(\widehat{Q}_{SS}\big)^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_S\big\|_\infty + \lambda_n |\!|\!|\big(\widehat{Q}_{SS}\big)^{-1}|\!|\!|_\infty, \tag{7.6}$$

(3) If $h \geq k$*, at least the smallest (in absolute value)* $p - h$ *entries in* S^c *are estimated exactly as zero and we have a simpler (possibly tighter) bound:*

$$\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty \leq \big\|\big(\widehat{Q}_{\widehat{U}\widehat{U}}\big)^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_{\widehat{U}}\big\|_\infty \tag{7.7}$$

where $\widehat{U}$ *is defined as the* h *largest absolute entries of* $\widetilde{\boldsymbol{\theta}}$ *including* S*.*

Remarks. The above theorem will be instantiated for the specific cases of sparse linear and sparse graphical models in subsequent corollaries (for which we will bound terms involving $\nabla\mathcal{L}(\boldsymbol{\theta}^*)$, $\widehat{\boldsymbol{z}}$ and $\widehat{Q}$). Though conditions (a) and (b) in Theorem 17 seem apparently more stringent than the case where $h = 0$ (vanilla Lasso), we will see in corollaries that they are uniformly upper bounded for all selections, under the asymptotically same probability as $h = 0$.

Note also that for $h = 0$, we recover the results for the vanilla ℓ_1 norm. Furthermore, by the statement (1) in the theorem, if $h < k$, $\widehat{U}$ only contains relevant feature indices and some relevant features are not penalized. If $h \geq k$, $\widehat{U}$ includes all relevant indices (and some non-relevant indices). In this case, the second term in (7.6) disappears, but the term $\big\|\big(\widehat{Q}_{\widehat{U}\widehat{U}}\big)^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_{\widehat{U}}\big\|_\infty$ increases as $|\widehat{U}|$ gets larger. Moreover, the condition that $n \asymp (k + h)\log p$ will be violated as h approaches p. While we do not know the true sparsity k a priori in many problems, we implicitly assume that we can set $h \asymp k$ (i.e., by cross-validation).

Now we turn to ℓ_2 bound under the same conditions:

Theorem 18. *Consider the problem with trimmed regularizer* (7.2) *where all conditions in Theorem 17 hold. Then, for any local minimum of* (7.2)*, the parameter estimation error in terms of ℓ_2 norm is upper bounded: for some constant C,*

$$\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2 \leq \begin{cases} C\lambda_n \left(\sqrt{k}/2 + \sqrt{k-h}\right) & \text{if } h < k \\ C\lambda_n \sqrt{h}/2 & \text{otherwise} \end{cases}$$

Remarks. The benefit of using trimmed ℓ_1 over standard ℓ_1 can be clearly seen in Theorem 18. Even though both have the same asymptotic convergence rates (in fact, standard ℓ_1 is already information theoretically optimal in many cases such as high-dimensional least squares), trimmed ℓ_1 has a smaller constant: $\frac{3C\lambda_n\sqrt{k}}{2}$ for standard ℓ_1 ($h = 0$) vs. $\frac{C\lambda_n\sqrt{k}}{2}$ for trimmed ℓ_1 ($h = k$). Comparing with non-convex (μ, γ)-amenable regularizers SCAD or MCP, we can also observe that the estimation bounds are asymptotically the same: $\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty \leq c\|(\widehat{Q}_{SS})^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_S\|_\infty$ and $\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2 \leq c\lambda_n\sqrt{k}$. However, the constant c here for those regularizers might be too large if μ is not small enough, since it involves $\frac{1}{\kappa_l - \mu}$ term (vs. $\frac{1}{\kappa_l}$ for the trimmed ℓ_1.) Moreover amenable non-convex regularizers require the additional constraint $\|\boldsymbol{\theta}\|_1 \leq R$ in their optimization problems for theoretical guarantees, along with further assumptions on $\boldsymbol{\theta}^*$ and tuning parameter R, and the true parameter must be feasible for their modified program (see [231]). The condition $\|\boldsymbol{\theta}^*\|_1 \leq R$ is stringent with respect to the analysis: as p and k increase, in order for R to remain constant, $\|\boldsymbol{\theta}^*\|_\infty$ must shrink to get satisfactory theoretical bounds. In contrast, while choosing the trimming parameter h requires cross-validation, it is possible to set h on a similar order as k.

We are now ready to apply our main theorem to the popular high-dimensional problems introduced in Section 7.2: sparse linear regression and sparse graphical model estimation. Due to space constraint, the results for sparse graphical models are provided in the supplementary materials.

7.3.1 Sparse Linear Regression

Motivated by the information theoretic bound for arbitrary methods, all previous analyses of sparse linear regression assume $n \geq c_0 k \log p$ for sufficiently large constant c_0. We also assume $n \geq c_0 \max\{k, h\} \log p$, provided $h \asymp k$.

Corollary 4. *Consider the model* (7.3) *where $\boldsymbol{\epsilon}$ is sub-Gaussian. Suppose we solve* (7.4) *with the selection of:*

(a) $\lambda_n \geq c_\ell \sqrt{\frac{\log p}{n}}$ for some constant c_ℓ depending only on the sub-Gaussian parameters of X and $\boldsymbol{\epsilon}$

(b) h satisfying: for any selection of $T \subseteq [p]$ s.t. $|T| = h$,

$$\left\| \left(\widehat{\Gamma}^{-1}\right)_{UU} \right\|_\infty \leq c_\infty, \qquad \left\| \widehat{\Gamma}_{U^cU} \left(\widehat{\Gamma}_{UU}\right)^{-1} \right\|_\infty \leq \eta,$$
$$\max\left\{ \lambda_{\max}(\widehat{\Gamma}_{U^cU^c}), \lambda_{\max}\big((\widehat{\Gamma}_{UU})^{-1}\big) \right\} \leq c_u \tag{7.8}$$

where $\widehat{\Gamma} = \frac{X^\top X}{n}$ is the sample covariance matrix and $\lambda_{\max}$ is the maximum singular value of a matrix.

Further suppose $\frac{1}{2}\boldsymbol{\theta}^*_{\min} \geq c_1\sqrt{\frac{\log p}{n}} + \lambda_n c_\infty$ for some constant c_1. Then with high probability at least $1 - c_2\exp(-c_3 \log p)$, any local minimum $\widetilde{\boldsymbol{\theta}}$ of (7.4) satisfies

(a) for every pair $j_1 \in S, j_2 \in S^c$, we have $|\widetilde{\boldsymbol{\theta}}_{j_1}| > |\widetilde{\boldsymbol{\theta}}_{j_2}|$,

(b) if $h < k$, all $j \in S^c$ are successfully estimated as zero and we have

$$\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty \leq c_1\sqrt{\frac{\log p}{n}} + \lambda_n c_\infty,$$
$$\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2 \leq c_4\sqrt{\frac{\log p}{n}}\left(\sqrt{k}/2 + \sqrt{k-h}\right).$$

(c) if $h \geq k$, at least the smallest $p - h$ entries in S^c have exactly zero and we have

$$\|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty \leq c_1\sqrt{\frac{\log p}{n}}, \quad \|\widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2 \leq \frac{c_4}{2}\sqrt{\frac{h\log p}{n}}.$$

Remarks. The conditions in Corollary 4 are also used in previous work and may be shown to hold with high probability via standard concentration bounds for sub-Gaussian matrices. In particular (7.8) is known as an incoherence condition for sparse least square estimators [350]. In the case of vanilla Lasso, estimation will fail if the incoherence condition is violated [350]. In contrast, we confirm by simulations in Section 7.4 that the trimmed ℓ_1 problem (7.4) can succeed even when this condition is not met. Therefore we conjecture that the incoherence condition could be relaxed in our case, similarly to the case of non-convex μ-amenable regularizers such as SCAD or MCP [231]. Proving this conjecture is highly non-trivial, since our penalty is based on a sum of absolute values, which is not μ-amenable; we leave the proof for future work.

We develop and analyze a block coordinate descent algorithm for solving objective (7.1), which is highly nonconvex problem because of the coupling of w and θ in the regularizer. The block-coordinate descent algorithm uses simple nonlinear operators:

$$\mathrm{proj}_{\mathcal{S}}(\boldsymbol{z}) := \arg\min_{\boldsymbol{w}\in\mathcal{S}} \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{w}\|^2$$

$$\mathrm{prox}_{\eta\lambda\mathcal{R}(\cdot,\boldsymbol{w}^{k+1})}(\boldsymbol{z}) := \arg\min_{\boldsymbol{\theta}} \frac{1}{2\eta\lambda}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2 + \sum_{j=1}^{p} w_j^{k+1}|\theta_j|$$

Algorithm 14 Block Coordinate Descent for (7.1)

Input: λ, η, and τ.
Initialize: $\boldsymbol{\theta}^0$, $\boldsymbol{w}^0$, and $k = 0$.
while not converged **do**
  $\boldsymbol{w}^{k+1} \leftarrow \text{proj}_{\mathcal{S}}[\boldsymbol{w}^k - \tau \boldsymbol{r}(\boldsymbol{\theta}^k)]$
  $\boldsymbol{\theta}^{k+1} \leftarrow \text{prox}_{\eta\lambda\mathcal{R}(\cdot,\boldsymbol{w}^{k+1})}[\boldsymbol{\theta}^k - \eta\nabla\mathcal{L}(\boldsymbol{\theta}^k)]$
  $k \leftarrow k + 1$
end while
Output: $\boldsymbol{\theta}^k$, $\boldsymbol{w}^k$.

Adding a block of weights $\boldsymbol{w}$ decouples the problem into simply computable pieces. Projection onto a polyhedral set is straightforward, while the prox operator is a weighted soft thresholding step.

We analyze Algorithm 7.1 using the structure of (7.1) instead of relying on the DC formulation for (7.2). The convergence analysis is summarized in Theorem 19 below. The analysis centers on the general objective function

$$\min_{\boldsymbol{\theta},\boldsymbol{w}} F(\boldsymbol{\theta}, \boldsymbol{w}) := \mathcal{L}(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{p} w_i r_i(\boldsymbol{\theta}) + \delta(\boldsymbol{w}|\mathcal{S}), \tag{7.9}$$

where $\delta(\boldsymbol{w}|\mathcal{S})$ enforces $w \in \mathcal{S}$. We let

$$\boldsymbol{r}(\boldsymbol{\theta}) = \begin{bmatrix} r_1(\boldsymbol{x}) & \dots & r_p(\boldsymbol{x}) \end{bmatrix}^T, \mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w}) = \langle \boldsymbol{w}, \boldsymbol{r}(\boldsymbol{\theta}) \rangle.$$

In the case of trimmed ℓ_1, r is the ℓ_1 norm, $r_i(x) = |x_i|$ and $\mathcal{S}$ encodes the constraints $0 \leq w_i \leq 1$, $\mathbf{1}^T\boldsymbol{w} = p - h$.

We make the following assumptions.

Assumption 5. *(a) $\mathcal{L}$ is a smooth closed convex function with an L_f-Lipchitz continuous gradient; (b) r_i are convex, and L_r-Lipchitz continuous and (c) $\mathcal{S}$ is a closed convex set and F is bounded below.*

In the non-convex setting, we do not have access to distances to optimal iterates or best function values, as we do for strongly convex and convex problems. Instead, we use distance to stationarity to analyze the algorithm. Objective (7.9) is highly non-convex, so we design a stationarity criterion, which goes to 0 as we approach stationary points. The analysis then shows Algorithm 7.1 drives this measure to 0, i.e. converges to stationarity. In our setting, every stationary point of (7.1) corresponds to a local optimum in $\boldsymbol{w}$ with $\boldsymbol{\theta}$ fixed, and a local optimum in $\boldsymbol{\theta}$ with $\boldsymbol{w}$ fixed.

Definition 7 (Stationarity)**.** *Define the stationarity condition $T(\boldsymbol{\theta}, \boldsymbol{w})$ by*

$$\begin{aligned} T(\boldsymbol{\theta}, \boldsymbol{w}) = \min\{\|\boldsymbol{u}\|^2 + \|\boldsymbol{v}\|^2 : &\boldsymbol{u} \in \partial_\theta F(\boldsymbol{\theta}, \boldsymbol{w}), \\ &\boldsymbol{v} \in \partial_w F(\boldsymbol{\theta}, \boldsymbol{w})\}. \end{aligned} \tag{7.10}$$

The pair $(\boldsymbol{\theta}, \boldsymbol{w})$ *is a stationary point when* $T(\boldsymbol{\theta}, \boldsymbol{w}) = 0$.

Theorem 19. *Suppose Assumptions 5 (a-c) hold, and define the quantity* $\mathcal{G}$ *as follows:*

$$\mathcal{G}_k := \frac{L_f}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \frac{\lambda}{\tau}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2.$$

With step size $\eta = 1/L_f$*, we have,*

$$\min_k \mathcal{G}_k \leq \frac{1}{K}\sum_{k=1}^{K} \mathcal{G}_k \leq \frac{1}{K}(F(\boldsymbol{\theta}^1) - F^*)$$
$$T(\boldsymbol{\theta}^{k+1}, \boldsymbol{w}^{k+1}) \leq (4 + 2\lambda L_r/L_f)\mathcal{G}_k,$$

and therefore

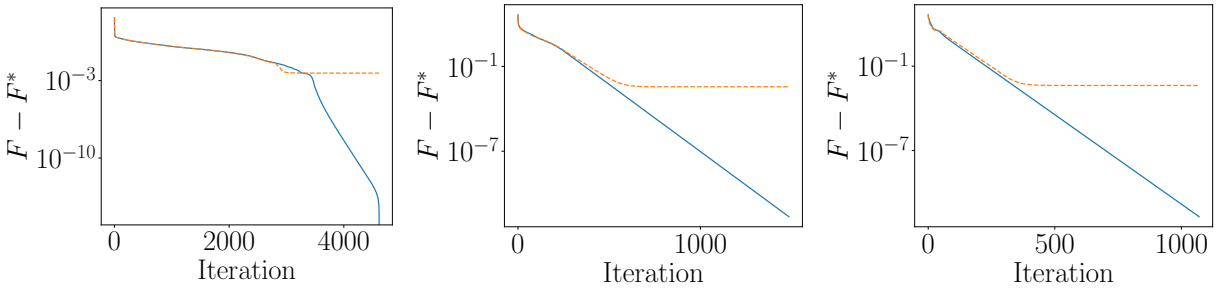
$$\min_{k=1:K}\{T(\boldsymbol{\theta}^k, \boldsymbol{w}^k)\} \leq \frac{4 + 2\lambda L_r/L_f}{K}(F(\boldsymbol{\theta}^1) - F^*).$$

The trimmed ℓ_1 problem satisfies Assumption 5 and hence Theorem 19 holds. Algorithm 14 for (7.1) converges at a sublinear rate measured using the distance to stationarity T (7.10), see Theorem 19. In the simulation experiments of Section 7.4, we will observe that the iterates converge to very close points regardless of initializations. [205] use similar concepts to analyze their DC-based algorithm, since it is also developed for a nonconvex model.

We include a small numerical experiment, comparing Algorithm 1 with Algorithm 2 of [205]. The authors proposed multiple approaches for DC programs; the prox-type algorithm (Algorithm 2) did particularly well for subset selection, see Figure 2 of [205]. We generate Lasso simulation data with variables of dimension 500, and 100 samples. The number of nonzero elements in the true generating variable is 10. We take $h = 25$, and apply both Algorithm 14 and Algorithm 2 of [205]. Initial progress of the methods is comparable, but Algorithm 14 continues at a linear rate to a lower value of the objective, while Algorithm 2 of [205] tapers off at a higher objective value. We consistently observe this phenomenon for a broad range of settings, regardless of hyperparameters; see convergence comparisons in Figure 7.1 for $\lambda \in \{0.5, 5, 20\}$. This comparison is very brief; we leave a detailed study comparing Algorithm 14 with DC-based algorithms to future algorithmic work, along with further analysis of Algorithm 14 and its variants under the Kurdyka-Lojasiewicz assumption [24].

7.4 Experimental Results

Simulations for sparse linear regression. We design four experiments. For all experiments except the third one where we investigate the effect of small regularization parameters, we choose the regularization parameters via cross-validation from the set: $\log_{10}\lambda \in \{-3.0, -2.8, \ldots, 1.0\}$. For non-convex penalties requiring additional parameter, we just fix



(a) $\lambda = 0.5$ (b) $\lambda = 5$ (c) $\lambda = 20$

Figure 7.1: Convergence of Algorithm 14 (blue solid) vs. Algorithm 2 of [205] (orange dot). We see consistent results across parameter settings.

their values (2.5 for MCP and 3.0 for SCAD respectively) since they are not sensitive to results. When we generate feature vectors, we consider two different covariance matrices of normal distribution as introduced in [231] to see how regularizers are affected by the incoherence condition.

In our first experiment, we generate i.i.d. observations from $x_i \sim N(0, M_2(\theta))$ where $M_2(\theta) = \theta \mathbf{1}\mathbf{1}^T + (1-\theta)I_p$ with $\theta = 0.7$.[1] This choice of $M_2(\theta)$ satisfies the incoherence condition [231]. We give non-zero values β^* with the magnitude sampled from $N(0, 5^2)$, at k random positions, and the response variables are generated by $y_i = x_i^T \beta^* + \epsilon_i$, where $\epsilon_i \sim N(0, 1^2)$. In Figure 7.2 (a) $\sim$ (c), we set $(p, k) = (128, 8), (256, 16), (512, 32)$ and increase the sample size n. The probability of correct support recovery for trimmed Lasso is higher than baselines for all samples in all cases. Figure 7.2(d) corroborates Corollary 4: any local optimum with trimmed ℓ_1 is close to points with correct support regardless of initialization; see comparisons against baselines with same setting in Figure 7.2(e).

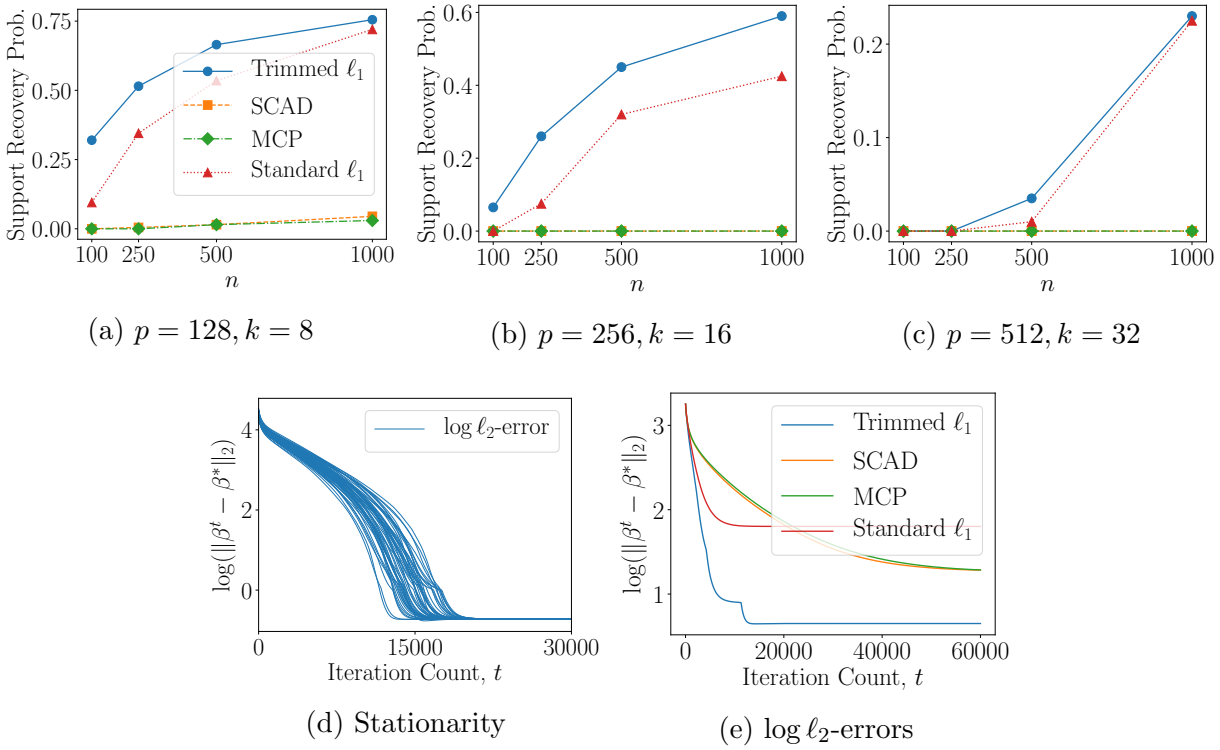
In the second experiment, we replace $M_2(\theta)$ with $M_1(\theta)$, which does not satisfy the incoherence condition.[2] Trimmed still outperforms comparison approaches (Figure 7.3). Lasso is omitted from Figure 7.3(e) as it always fails in this setting.

Our next experiment compares Trimmed Lasso against vanilla Lasso where both λ and true non-zeros are small: $\log \lambda \in \{-3.0, -2.8, \ldots, -1.0\}$ and $\beta^* \sim N(0, 0.8^2)$. When the magnitude of $\boldsymbol{\theta}^*$ is large, standard Lasso tends to choose a small value of λ to reduce the bias of the estimate while Trimmed Lasso gives good performance even for large values of λ as long as h is chosen suitably. Figure 7.4(a) also confirms the superiority of Trimmed Lasso in a small regime of λ with a proper choice of h.

---

[1] M_1 and M_2 as defined in [231].

[2] $M_1(\theta)$ is a matrix with 1's on the diagonal, θ's in the first k positions of the $(k+1)^{\text{st}}$ row and column, and 0's elsewhere.



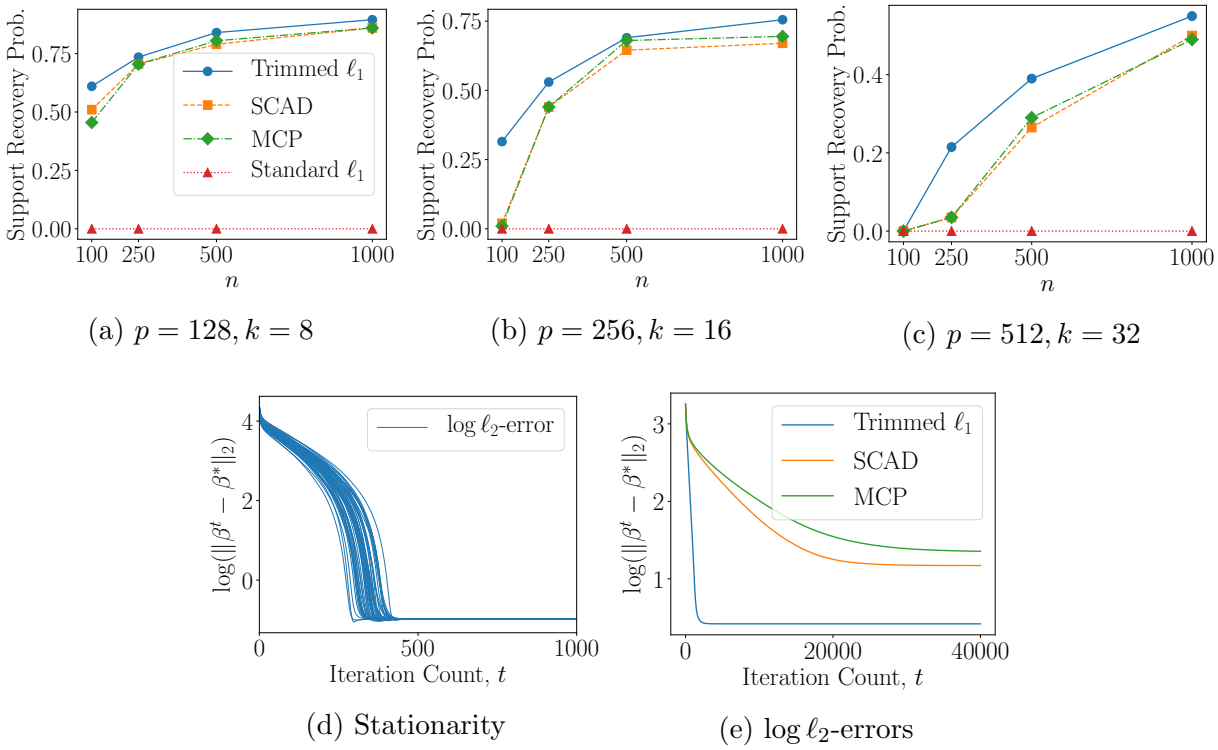
(a) $p = 128, k = 8$ (b) $p = 256, k = 16$ (c) $p = 512, k = 32$

(d) Stationarity (e) $\log \ell_2$-errors

Figure 7.2: Results for the incoherent case of the first experiments. **(a)∼(c)**: Probability of sucessful support recovery for Trimmed ℓ_1, SCAD, MCP, and standard ℓ_1 as sample size n increases. For **(d)**, **(e)**, we adopt the high-dimensional setting with $(n, p, k) = (160, 256, 16)$, and use 50 random initializations.

In the last experiment, we investigate the effect of choosing the trimming parameter h. Figure 7.4(b) and (c) show that Trimmed ℓ_1 outperforms if we set $h = k$ (note $(p-h)/p \approx 0.94$). As $h \downarrow 0$ (when $(p-h)/p = 1$), the performance approaches that of Lasso, as we can see in Corollary 4. Additional experiments on sparse Gaussian Graphical Models are provided as supplementary materials.

Input Structure Recovery of Compact Neural Networks. We apply the Trimmed ℓ_1 regularizer to recover input structures of deep models. We follow [271] and consider the regression model $y_i = \mathbf{1}^T \sigma(\boldsymbol{W}^* \boldsymbol{x}_i)$ with input dimension $p = 80$, hidden dimension $z = 20$, and ReLU activation $\sigma(\cdot)$. We generate i.i.d. data $\boldsymbol{x}_i \sim N(0, I_p)$ and $\boldsymbol{W}^* \in \mathbb{R}^{z \times p}$ such that ith row has exactly 4 non-zero entries from $N(0, \frac{p}{4z})$ to ensure that $\mathbb{E}[\|\boldsymbol{W}^* \boldsymbol{x}\|_{\ell_2}^2] = \|\boldsymbol{x}\|_{\ell_2}^2$ at only $4(i-1)+1 \sim 4i$ positions. For ℓ_0 and ℓ_1 regularizations, we optimize $\boldsymbol{W}$ using a projected gradient descent with prior knowledge of $\|\boldsymbol{W}^*\|_0$ and $\|\boldsymbol{W}^*\|_1$, and we use Algorithm 14 for



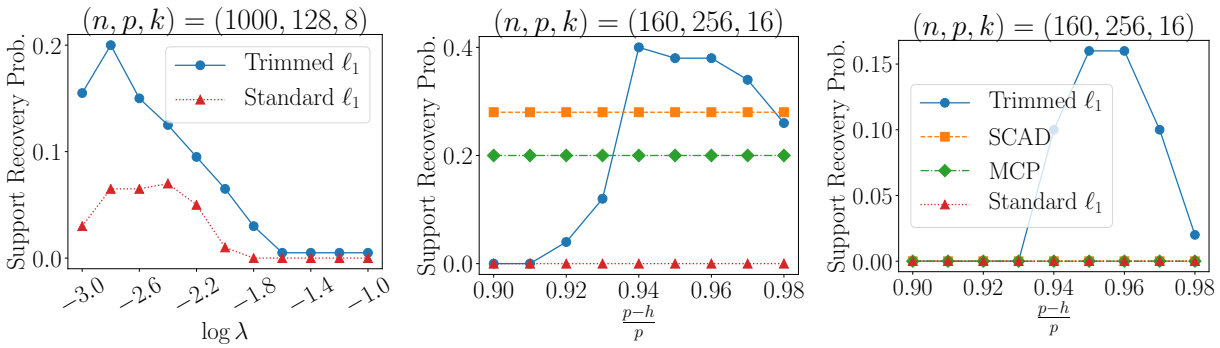
(a) $p = 128, k = 8$ (b) $p = 256, k = 16$ (c) $p = 512, k = 32$

(d) Stationarity (e) $\log \ell_2$-errors

Figure 7.3: Results for the non-incoherent case. **(a)∼(e)**: same as Figure 7.2.

trimmed ℓ_1 regularization with $h = 4z$ and $(\lambda, \tau) = (0.01, 0.1)$ obtained by cross-validation. We set the step size $\eta = 0.1$ for all approaches. We consider two sets of simulations with varying sample size n where the initial $\boldsymbol{W}_0$ is selected as (a) a small perturbation of $\boldsymbol{W}^*$ and (b) at random, as in [271]. Figure 7.7 shows the results where black dots indicate nonzero values in the weight matrix, and we can confirm that Trimmed ℓ_1 outperforms alternatives in terms of support recovery for both cases.

Pruning Deep Neural Networks. Several recent studies have shown that neural networks are highly over-parameterized, and we can prune the weight parameters/neurons with marginal effect on performance. Toward this, we consider trimmed regularization based network pruning. Suppose we have deep neural networks with L hidden layers. Let n_i be the number of neurons in the layer $\boldsymbol{h}_i$. The parameters we are interested in are $\mathcal{W} \coloneqq \{\boldsymbol{\theta}_l, \boldsymbol{b}_l\}_{l=1}^{L+1}$ for $\boldsymbol{\theta}_l \in \mathbb{R}^{n_{l-1} \times n_l}$ and $\boldsymbol{b}_l \in \mathbb{R}^{n_l}$ where $\boldsymbol{h}_0$ is the input feature $\boldsymbol{x}$ and $\boldsymbol{h}_{L+1}$ is the output $\boldsymbol{y}$. Then, for $l = 1, \ldots, L$, $\boldsymbol{h}_l = \mathrm{ReLU}(\boldsymbol{h}_{l-1}\boldsymbol{\theta}_l + \boldsymbol{b}_l)$. Since the edge-wise pruning will not give actual benefit in terms of computation, we prune unnecessary *neurons* through group-sparse encouraging regularizers. Specifically, given the weight parameter $\boldsymbol{\theta} \coloneqq \boldsymbol{\theta}_l$ between $\boldsymbol{h}_{l-1}$ and



(a) Small Regime (b) Non-incoherent (c) Incoherent

Figure 7.4: Plots for third and last experiments. **(a)**: Trimmed Lasso versus standard one in a small regime. We set $h = \lceil 0.05p \rceil$. **(b)**, **(c)**: Performance of the trimmed Lasso as the value of h varies.

$\boldsymbol{h}_l$, we consider the group norm extension of trimmed ℓ_1:

$$\mathcal{R}_l(\boldsymbol{\theta}, \boldsymbol{w}) \coloneqq \lambda \sum_{j=1}^{n_{l-1}} w_j \sqrt{\theta_{j,1}^2 + \cdots + \theta_{j,n_l}^2}$$

with the constraint of $\mathbf{1}^T \boldsymbol{w} = n_{l-1} - h$. Moreover, we can naturally make an extension to a convolutional layer with encouraging *activation map sparsity* as follows. If $\boldsymbol{\theta}$ is a weight parameter for 2-dimensional convolutional layer (most generally used) with $\boldsymbol{\theta} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times H \times W}$, the trimmed regularization term that induces activation map-wise sparsity is given by

$$\mathcal{R}_l(\boldsymbol{\theta}, \boldsymbol{w}) \coloneqq \lambda \sum_{j=1}^{C_{\text{out}}} w_j \sqrt{\sum_{m,n,k} \theta_{j,m,n,k}^2}$$

for all possible indices (m, n, k). Finally, we add all penalizing terms to a loss function to have

$$\mathcal{L}(\mathcal{W}; \mathcal{D}) + \sum_{l=1}^{L+1} \lambda_l \mathcal{R}_l(\boldsymbol{\theta}_l, \boldsymbol{w}_l)$$

where we allow different hyperparameters λ_l and h_l for each layer.

In Table 7.1, we compare trimmed group ℓ_1 regularization against vanilla group ℓ_1 on MNIST dataset using LeNet-300-100 architecture [217]. Here, we set the trimming parameter h to half sparsity level of the original model. For the vanilla group ℓ_1, we need larger λ values to obtain sparser models, for which we pay a significant loss of accuracy. In contrast, we can control the sparsity level using trimming parameters h with little or no drop of accuracy.

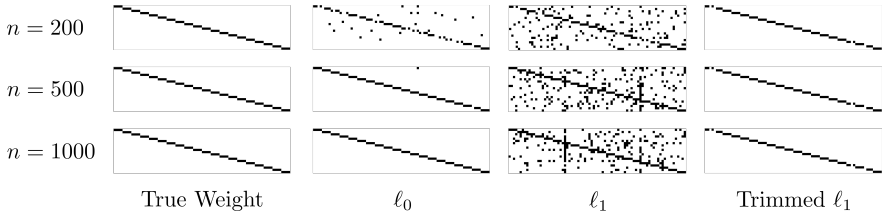


Figure 7.5:

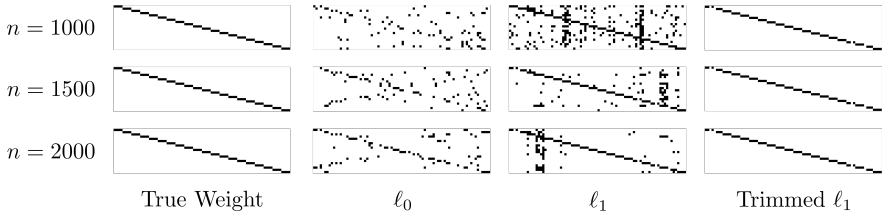


Figure 7.6:

Figure 7.7: Results for sparsity pattern recovery of deep models.

Table 7.1: Results on MNIST using LeNet-300-100.

Method	Pruned Model	Error (%)
No Regularization	784-300-100	**1.6**
grp ℓ_1	784-241-67	1.7
grp $\ell_{1_{\text{trim}}}$, h = half of original	**392-150-50**	**1.6**

Most algorithms for network pruning recently proposed are based on a variational Bayesian approach [104, 233]. Motivated by learning sparse structures via smoothed version of ℓ_0 norm [233], we propose a Bayesian neural network with trimmed regularization where we regard only $\boldsymbol{\theta}$ as Bayesian. Inspired by a relation between variational dropout and Bayesian neural networks [206], we specifically choose a fully factorized Gaussian as a variational distribution, $q_{\boldsymbol{\phi},\boldsymbol{\alpha}}(\theta_{i,j}) = \mathcal{N}(\phi_{i,j}, \alpha_{i,j}\phi_{i,j}^2)$, to approximate the true posterior and leave $\boldsymbol{w}$ to directly learn sparsity patterns. Then the problem is cast to maximizing corresponding evidence lower bound (ELBO),

$$\mathbb{E}_{q_{\boldsymbol{\phi},\boldsymbol{\alpha}}}[\mathcal{L}(\mathcal{W};\mathcal{D})] - \mathbb{KL}\big(q_{\boldsymbol{\phi},\boldsymbol{\alpha}}(\mathcal{W}) \| p(\mathcal{W})\big).$$

Table 7.2: Results on MNIST classification for LeNet 300-100 with Bayesian approaches. $h = \circ$ means that the trimming parameter h is set to the same sparsity level of $\circ$, and λ sep. indicates that different λ values are employed on each layer.

Method	Pruned Model	Error (%)
ℓ_0 [233]	219-214-100	**1.4**
ℓ_0, λ sep. [233]	266-88-33	1.8
Bayes grp $\ell_{1_{\text{trim}}}$, $h = \ell_0$	219-214-100	**1.4**
Bayes grp $\ell_{1_{\text{trim}}}$, $h = \ell_0$, λ sep.	266-88-33	1.6
Bayes grp $\ell_{1_{\text{trim}}}$, $h < \ell_0$, λ sep.	**245-75-25**	1.7

Table 7.3: Results on MNIST classification for LeNet-5-Caffe with Bayesian approaches.

Method	Pruned Model	Error (%)
ℓ_0 [233]	20-25-45-462	**0.9**
ℓ_0, λ sep. [233]	9-18-65-25	1.0
Bayes grp $\ell_{1_{\text{trim}}}$, $h < \ell_0$	20-25-45-150	**0.9**
Bayes grp $\ell_{1_{\text{trim}}}$, $h = \ell_0$, λ sep.	9-18-65-25	1.0
Bayes grp $\ell_{1_{\text{trim}}}$, $h < \ell_0$, λ sep.	**8-17-53-19**	1.0

Combined with trimmed ℓ_1 regularization, the objective is

$$\mathbb{E}_{q_{\boldsymbol{\phi},\boldsymbol{\alpha}}(\boldsymbol{\theta})}\Big[- \mathcal{L}(\mathcal{W};\mathcal{D}) + \sum_{l=1}^{L+1} \lambda_l \mathcal{R}_l(\boldsymbol{\theta}_l, \boldsymbol{w}_l)\Big] + \mathbb{KL}(q_{\boldsymbol{\phi},\boldsymbol{\alpha}}(\mathcal{W}) \| p(\mathcal{W})) \tag{7.11}$$

which can be interpreted as a sum of expected loss and expected trimmed group ℓ_1 penalizing term. [207] provide the efficient unbiased estimator of stochastic gradients for training $(\boldsymbol{\phi}, \boldsymbol{\alpha})$, via the reparameterization trick to avoid computing gradient of sampling process. In order to speed up our method, we approximate expected loss term in (7.11) using a local reparameterization trick [206] while the standard reparameterization trick is used for the penalty term.

Trimmed group ℓ_1 regularized Bayesian neural networks have smaller capacity with less error than other baselines (Table 7.2). Our model has lower error rate and better sparsity

even for convolutional network, LeNet-5-Caffe[3] (Table 7.3).[4]

The code is available at `https://github.com/abcdxyzpqrst/Trimmed_Penalty`.

7.5 Concluding Remarks

In this work we studied statistical properties of high-dimensional M-estimators with the trimmed ℓ_1 penalty, and demonstrated the value of trimmed regularization compared to convex and non-convex alternatives. We developed a provably convergent algorithm for the trimmed problem, based on specific problem structure rather than generic DC structure, with promising numerical results. A detailed comparison to DC based approaches is left to future work. Going beyond M-estimation, we showed that trimmed regularization can be beneficial for two deep learning tasks: input structure recovery and network pruning. As future work we plan to study trimming of general decomposable regularizers, including ℓ_1/ℓ_q norms, and further investigate the use of trimmed regularization in deep models.

7.6 Appendix

7.6.1 Sparse graphical models

We derive a corollary for the trimmed Graphical lasso:

$$\underset{\boldsymbol{\Theta}\in\mathcal{S}^p_{++}}{\text{minimize}}\ \operatorname{trace}\big(\widehat{\Sigma}\boldsymbol{\Theta}\big) - \log\det\big(\boldsymbol{\Theta}\big) + \lambda_n\mathcal{R}(\boldsymbol{\Theta}_{\text{off}}; h). \tag{7.12}$$

Following the strategy of [231], we assume that the sample size scales with the row sparsity d of true inverse covariance $\boldsymbol{\Theta}^* = (Cov(X))^{-1}$, which is a milder condition than other works (n scaling with k, the number of non zero entries of $\boldsymbol{\Theta}^*$):

Corollary 5. *Consider the program* (7.12) *where the x_i's are drawn from a sub-Gaussian and sample size $n > c_0 d^2 \log p$ with the selection of*

(a) $\lambda_n \geq c_\ell\sqrt{\frac{\log p}{n}}$ for some constant c_ℓ depending only on $\boldsymbol{\Theta}^*$

(b) h satisfying: for any selection of $T \subseteq \{1,2,\ldots,p\}\times\{1,2,\ldots,p\}$ s.t. $|T| = h$,

$$\begin{aligned}
&\Big\|\big(\boldsymbol{\Theta}^*\otimes\boldsymbol{\Theta}^*\big)_{UU}\Big\|_\infty \leq c_\infty, \\
&\max\Big\{\|\widehat{\Gamma}_{U^cU^c}\|_\infty, \|(\widehat{\Gamma}_{UU})^{-1}\|_\infty\Big\} \leq c_u \quad \textit{and} \\
&\Big\|\big(\boldsymbol{\Theta}^{*-1}\otimes\boldsymbol{\Theta}^{*-1}\big)_{U^cU}\Big(\big(\boldsymbol{\Theta}^{*-1}\otimes\boldsymbol{\Theta}^{*-1}\big)_{UU}\Big)^{-1}\Big\|_\infty \leq \eta.
\end{aligned} \tag{7.13}$$

[3] `https://github.com/BVLC/caffe/tree/master/examples/mnist`

[4] We only consider methods based on sparsity encouraging regularizers. State-of-the-art VIBNet [104] exploits the mutual information between each layer.

Further suppose that $\frac{1}{2}\mathbf{\Theta}^*_{\min}$ *is lower bounded by* $c_1\sqrt{\frac{\log p}{n}}+2\lambda_n c_\infty$ *for some constant* c_1*. Then with high probability at least* $1-c_2\exp(-c_3\log p)$*, any local minimum* $\widetilde{\mathbf{\Theta}}$ *of* (7.4) *has the following property:*

(a) For every pair $j_1 \in S, j_2 \in S^c$*,* $|\widetilde{\mathbf{\Theta}}_{j_1}| > |\widetilde{\mathbf{\Theta}}_{j_2}|$*,*

(b) If $h < k$*, all* $j \in S^c$ *are successfully estimated as zero and we have*

$$\|\widetilde{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_\infty \leq c_1\sqrt{\frac{\log p}{n}} + 2\lambda_n c_\infty \tag{7.14}$$

(c) If $h \geq k$*, at least the smallest* $p-h$ *entries in* S^c *have exactly zero and we have*

$$\|\widetilde{\mathbf{\Theta}} - \mathbf{\Theta}^*\|_\infty \leq c_1\sqrt{\frac{\log p}{n}}. \tag{7.15}$$

Note that condition (7.13) is the incoherence condition studied in [287], and the same remarks as those for sparse linear models (see Section 7.3.1) can be made.

7.6.2 Proofs

Proof of Theorem 17

We extend the standard PDW technique [349, 365, 231] for the trimmed regularizers. For any **fixed** T, we construct a primal and dual witness pair with the strict dual feasibility. Specifically, given the fixed T, consider the following program:

$$\underset{\boldsymbol{\theta}\in\Omega}{\text{minimize}} \;\; \mathcal{L}(\boldsymbol{\theta};\mathcal{D}) + \lambda_n \sum_{j\in T^c} |\theta_j|. \tag{7.16}$$

Note that the program (7.16) is convex (under (C-1)) where the regularizer is only effective over entries in (fixed) T^c. We construct the primal and dual pair $(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{z}})$ by the following restricted program

$$\widehat{\boldsymbol{\theta}} \in \underset{\boldsymbol{\theta}\in\mathbb{R}^U:\; \boldsymbol{\theta}\in\Omega}{\operatorname{argmin}} \;\; \mathcal{L}(\boldsymbol{\theta}) + \lambda_n \mathcal{R}(\boldsymbol{\theta}; h) \tag{7.17}$$

and (7.5). The following lemma can guarantee under the strict dual feasibility that any solution of (7.16) has the same sparsity structure on T^c with $\widehat{\boldsymbol{\theta}}$. Moreover, since the restricted program (7.5) is strictly convex as shown in the lemma below, we can conclude that $\widehat{\boldsymbol{\theta}}$ is the unique minimum point of the restricted program (7.16) given T.

Lemma 9. *Suppose that there exists a primal optimal solution* $\widehat{\boldsymbol{\theta}}$ *for* (7.16) *with associated sub-gradient (or dual)* $\widehat{\boldsymbol{z}}$ *such that* $\|\widehat{\boldsymbol{z}}_{U^c}\|_\infty < 1$*. Then any optimal solution* $\widetilde{\boldsymbol{\theta}}$ *of* (7.16) *will satisfy* $\widetilde{\boldsymbol{\theta}}_j = 0$ *for all* $j \in U^c$*.*

Proof. The lemma can be directly achieved by the basic property of convex optimization problem, as developed in existing works using PDW [349, 365]. Note that even though the original problem with the trimmed regularizer is not convex, (7.16) given T is convex. Therefore, by complementary slackness, we have $\sum_{j\in T^c} |\widetilde{\boldsymbol{\theta}}_j| = \langle \widehat{\boldsymbol{z}}_{T^c}, \widetilde{\boldsymbol{\theta}}_{T^c}\rangle$. Therefore, any optimal solution of (7.16) will satisfy $\widetilde{\boldsymbol{\theta}}_j = 0$ for all $j \in U^c$ since the associated (absolute) sub-gradient is strictly smaller than 1 by the assumption in the statement. □

Lemma 10 (Section A.2 of [231])**.** *Under* (C-2)*, the loss function $\mathcal{L}(\boldsymbol{\theta})$ is strictly convex on $\boldsymbol{\theta} \in \mathbb{R}^U$ and hence $\big(\nabla^2\mathcal{L}(\boldsymbol{\theta})\big)_{UU}$ is invertible if $n \geq \frac{2\tau_1}{\kappa_l}(k+h)\log p$.*

Now from the definition of $\widehat{Q}$, we have

$$\widehat{Q}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) = \nabla\mathcal{L}(\widehat{\boldsymbol{\theta}}) - \nabla\mathcal{L}(\boldsymbol{\theta}^*) \tag{7.18}$$

where $\widehat{Q}$ is decomposed as $\begin{bmatrix} \widehat{Q}_{UU} & \widehat{Q}_{UU^c} \\ \widehat{Q}_{U^cU} & \widehat{Q}_{U^cU^c} \end{bmatrix}$. Then by the invertibility of $\big(\nabla^2\mathcal{L}(\boldsymbol{\theta})\big)_{UU}$ in Lemma 10 and the zero sub-gradient condition in (7.5) we have

$$\widehat{\boldsymbol{\theta}}_U - \boldsymbol{\theta}^*_U = \Big(\widehat{Q}_{UU}\Big)^{-1}\Big(-\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U - \lambda_n\widehat{\boldsymbol{z}}_U\Big). \tag{7.19}$$

Since both $\widehat{\boldsymbol{\theta}}_{U^c}$ and $\boldsymbol{\theta}^*_{U^c}$ are zero vectors, we obtain

$$\begin{aligned} \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty &= \Big\|\Big(\widehat{Q}_{UU}\Big)^{-1}\Big(-\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U - \lambda_n\widehat{\boldsymbol{z}}_U\Big)\Big\|_\infty \\ &\leq \Big\|\Big(\widehat{Q}_{UU}\Big)^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U\Big\|_\infty + \lambda_n\Big|\Big|\Big|\big(\widehat{Q}_{UU}\big)^{-1}\Big|\Big|\Big|_\infty. \end{aligned} \tag{7.20}$$

Therefore, under the assumption on $\boldsymbol{\theta}^*_{\min}$ in the statement, the selection of T in which there exists some (j, j') s.t. $j \in S$, $j \in T^c$, $j' \in S^c$ and $j' \in T$, yields contradictory solution with (7.2). Under the strict dual feasibility condition for this specific choice of T (along with Lemma 10) can guarantee that there is no local minimum for that choice of T. Hence, (7.21) can guarantee that for every pair (j_1, j_2) such that $j_1 \in S$ and $j_2 \notin S$, we have $|\widetilde{\boldsymbol{\theta}}_{j_1}| > |\widetilde{\boldsymbol{\theta}}_{j_2}|$ (since $\widetilde{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}$). Note that for any *valid* selection of T, this statement holds. This immediately implies that any local minimum of (7.2) satisfies this property as well, as in the statement.

Finally turning to the bound when $h \geq k$, we have $U = T$ since all entries in S are not penalized as shown above. In this case, $\widehat{\boldsymbol{z}}_U$ becomes zero vector (since V is empty in the construction of $\widehat{\boldsymbol{z}}$), and the bound in (7.21) will be tighter as

$$\begin{aligned} \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_\infty &= \Big\|\Big(\widehat{Q}_{UU}\Big)^{-1}\Big(-\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U - \lambda_n\widehat{\boldsymbol{z}}_U\Big)\Big\|_\infty \\ &\leq \Big\|\Big(\widehat{Q}_{UU}\Big)^{-1}\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U\Big\|_\infty, \end{aligned} \tag{7.21}$$

as claimed.

Proof of Theorem 18

Here we adopt the strategy developed in [366] for analyzing local optima of trimmed loss function. Since our loss function $\mathcal{L}$ is convex, the story derived in this subsection can also be applied to results of [257]. However, in order to simplify the procedure, we will not utilize the convexity of $\mathcal{L}$ and instead place the side constraint $\|\boldsymbol{\theta}\|_1 \leq R$ and some additional assumptions (see [229] for details). As in [366], we introduce the the shorthand to denote local optimal error vector: $\widetilde{\Delta} := \widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$ given an *arbitrary* local minimum $(\widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{w}})$ of (7.2). We additionally define H to denote the set of indices not penalized by $\widetilde{\boldsymbol{w}}$ (that is, $\widetilde{\boldsymbol{w}}_j = 0$ for $j \in H$, $\widetilde{\boldsymbol{w}}_j = 1$ for $j \in H^c$ and $|H| = h$). Utilizing the fact that $(\widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{w}})$ is a local minimum of (7.2), we have an inequality

$$\left\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}\big(\boldsymbol{\theta}^* + \widetilde{\Delta}\big),\, \widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta} \right\rangle \leq -\left\langle \partial \lambda \mathcal{R}(\boldsymbol{\theta}^* + \widetilde{\Delta}; h),\, \widetilde{\boldsymbol{\theta}} - \boldsymbol{\theta} \right\rangle \quad \text{for any feasible } \boldsymbol{\theta}.$$

This inequality comes from the first order stationary condition (see [230] for details) in terms of $\boldsymbol{\theta}$ fixing $\boldsymbol{w}$ at $\widetilde{\boldsymbol{w}}$. Here, if we take $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ above, we have

$$\left\langle \nabla \mathcal{L}\big(\boldsymbol{\theta}^* + \widetilde{\Delta}\big),\, \widetilde{\Delta} \right\rangle \leq -\left\langle \partial \lambda \mathcal{R}(\boldsymbol{\theta}^* + \widetilde{\Delta}; h),\, \widetilde{\Delta} \right\rangle \overset{(i)}{\leq} \lambda(\|\boldsymbol{\theta}^*_{H^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1)$$

where S is true support set of $\boldsymbol{\theta}^*$ and the inequality (i) holds due to the convexity of ℓ_1 norm.

i) $h < k$**:** By Theorem 17, we can guarantee with high probability that $H \subset S$. Then, by triangular inequality (in inequality (ii) below) and the fact that $\boldsymbol{\theta}^*$ is S-sparse vector, we have

$$\begin{aligned}
\left\langle \nabla \mathcal{L}\big(\boldsymbol{\theta}^* + \widetilde{\Delta}\big),\, \widetilde{\Delta} \right\rangle &\leq \lambda(\|\boldsymbol{\theta}^*_{H^c}\|_1 + \|\widetilde{\Delta}_{S^c}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1) = \lambda(\|\boldsymbol{\theta}^*_{H^c} + \widetilde{\Delta}_{S^c}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1) \\
&\overset{(ii)}{\leq} \lambda\big(\|\boldsymbol{\theta}^*_{H^c} + \widetilde{\Delta}_{S^c} + \widetilde{\Delta}_{S-H}\|_1 + \|\widetilde{\Delta}_{S-H}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1\big) = \lambda(\|\widetilde{\Delta}_{S-H}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1)\,.
\end{aligned} \tag{7.22}$$

Combining (7.22) and (C-2) yields

$$\kappa_l \|\widetilde{\Delta}\|_2^2 - \tau_1 \frac{\log p}{n} \|\widetilde{\Delta}\|_1^2 \leq \left\langle \nabla \mathcal{L}\big(\boldsymbol{\theta}^* + \widetilde{\Delta}\big) - \nabla \mathcal{L}\big(\boldsymbol{\theta}^*\big),\, \widetilde{\Delta} \right\rangle \leq -\left\langle \nabla \mathcal{L}\big(\boldsymbol{\theta}^*\big),\, \widetilde{\Delta} \right\rangle + \lambda\left(\|\widetilde{\Delta}_{S-H}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1\right).$$

If we assume $\max\left\{\|\nabla \mathcal{L}\big(\boldsymbol{\theta}^*\big)\|_\infty, 2\rho\tau_1 \frac{\log p}{n}\right\} \leq \frac{\lambda}{4}$ (which are slightly different to assumptions in the statement, however they are purely for simplicity and can be relaxed if we use the convexity of $\mathcal{L}$, as we mentioned in the beginning of the proof), we can conclude that

$$\begin{aligned}
0 \leq \kappa_l \|\widetilde{\Delta}\|_2^2 &\leq \left\|\nabla \mathcal{L}\big(\boldsymbol{\theta}^*\big)\right\|_\infty \|\widetilde{\Delta}\|_1 + \lambda\left(\|\widetilde{\Delta}_{S-H}\|_1 - \|\widetilde{\Delta}_{S^c}\|_1\right) + 2\rho\tau_1 \frac{\log p}{n} \|\widetilde{\Delta}\|_1 \\
&\leq \frac{\lambda}{2}\|\widetilde{\Delta}\|_1 - \lambda\|\widetilde{\Delta}_{S^c}\|_1 + \lambda\|\widetilde{\Delta}_{S-H}\|_1 \leq \frac{\lambda}{2}\|\widetilde{\Delta}_S\|_1 - \frac{\lambda}{2}\|\widetilde{\Delta}_{S^c}\|_1 + \lambda\|\widetilde{\Delta}_{S-H}\|_2 \leq \frac{\lambda}{2}\|\widetilde{\Delta}_S\|_1 + \lambda\|\widetilde{\Delta}_{S-H}\|_2.
\end{aligned} \tag{7.23}$$

As a result, we can finally have an ℓ_2 error bound as follows:

$$\kappa_l\|\widetilde{\Delta}\|_2^2 \leq \frac{\lambda\sqrt{k}}{2}\|\widetilde{\Delta}_S\|_2 + \lambda\sqrt{k-h}\|\widetilde{\Delta}_{S-H}\|_2 \leq \left(\frac{\lambda\sqrt{k}}{2} + \lambda\sqrt{k-h}\right)\|\widetilde{\Delta}\|_2$$

implying that

$$\|\widetilde{\Delta}\|_2 \leq \frac{1}{\kappa_l}\left(\frac{\lambda\sqrt{k}}{2} + \lambda\sqrt{k-h}\right).$$

ii) $h \geq k$**:** As in the previous case, Theorem 17 can guarantee $S \subseteq H$ where equality holds if $h = k$. Instead of (7.22), now we have

$$\begin{aligned}\left\langle \nabla\mathcal{L}\big(\boldsymbol{\theta}^* + \widetilde{\Delta}\big),\, \widetilde{\Delta}\right\rangle &\leq \lambda(\|\boldsymbol{\theta}^*_{H^c}\|_1 + \|\widetilde{\Delta}_{H^c}\|_1 - \|\widetilde{\Delta}_{H^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1) \\ = \lambda(\|\boldsymbol{\theta}^*_{H^c} + \widetilde{\Delta}_{H^c}\|_1 - \|\widetilde{\Delta}_{H^c}\|_1 - \|\widetilde{\boldsymbol{\theta}}_{H^c}\|_1) &= -\|\widetilde{\Delta}_{H^c}\|_1. \end{aligned} \tag{7.24}$$

By similar reasoning in the case of i), we combine (7.24) and (C-2) to obtain

$$0 \leq \kappa_l\|\widetilde{\Delta}\|_2^2 \leq \frac{\lambda}{2}\|\widetilde{\Delta}\|_1 - \lambda\|\widetilde{\Delta}_{H^c}\|_1 \leq \frac{\lambda}{2}\|\widetilde{\Delta}_H\|_1 - \frac{\lambda}{2}\|\widetilde{\Delta}_{H^c}\|_1 \leq \frac{\lambda}{2}\|\widetilde{\Delta}_H\|_1 \leq \frac{\lambda\sqrt{h}}{2}\|\widetilde{\Delta}\|_2 \tag{7.25}$$

implying that

$$\|\widetilde{\Delta}\|_2 \leq \frac{1}{\kappa_l}\frac{\lambda\sqrt{h}}{2}.$$

Proof of Corollary 4

The proof our corollary is similar to that of Corollary 1 of [231], who derive the result for (μ, γ)-amenable regularizers. Here we only describe the parts that need to be modified from [231].

In order to utilize theorems in the main paper, we need to establish the RSC condition (C-2) and the strict dual feasibility: $\|\widehat{\boldsymbol{z}}_{U^c}\|_\infty \leq 1 - \delta$

First, the RSC is known to hold w.h.p as shown in several previous works such as Lemma 11.

Lemma 11 (Corollary 1 of [230])**.** *The RSC condition in* (C-2) *for linear models holds with high probability with* $\kappa_l = \frac{1}{2}\lambda_{\min}(\Sigma_x)$ *and* $\tau_1 \asymp 1$*, under sub-Gaussian assumptions in the statement.*

In order to show the remaining strict dual feasibility condition of our PDW construction, we consider (7.18) (by the zero-subgradient and the definition of $\widehat{Q}$) in the block form:

$$\begin{bmatrix} \widehat{Q}_{TT} & \widehat{Q}_{TV} & \widehat{Q}_{TU^c} \\ \widehat{Q}_{VT} & \widehat{Q}_{VV} & \widehat{Q}_{VU^c} \\ \widehat{Q}_{U^cT} & \widehat{Q}_{U^cV} & \widehat{Q}_{U^cU^c} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*_T \\ \widehat{\boldsymbol{\theta}}_V - \boldsymbol{\theta}^*_V \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \nabla\mathcal{L}(\boldsymbol{\theta}^*)_T \\ \nabla\mathcal{L}(\boldsymbol{\theta}^*)_V \\ \nabla\mathcal{L}(\boldsymbol{\theta}^*)_{U^c} \end{bmatrix} + \lambda_n \begin{bmatrix} \mathbf{0} \\ \widehat{\boldsymbol{z}}_V \\ \widehat{\boldsymbol{z}}_{U^c} \end{bmatrix} = \mathbf{0}. \tag{7.26}$$

By simple manipulation, we can obtain

$$\widehat{\boldsymbol{z}}_{U^c} = \frac{1}{\lambda_n}\left\{-\nabla\mathcal{L}(\boldsymbol{\theta}^*)_{U^c} + \widehat{Q}_{U^cU}\Big(\widehat{Q}_{UU}\Big)^{-1}\Big(-\nabla\mathcal{L}(\boldsymbol{\theta}^*)_U - \lambda_n\widehat{\boldsymbol{z}}_V\Big)\right\}. \tag{7.27}$$

Here note that our construction of PDW can guarantee the ℓ_∞ bound in (7.21). In case of (7.4), since we have $\nabla\mathcal{L}(\boldsymbol{\theta}) = \widehat{\Gamma}\boldsymbol{\theta} - \widehat{\gamma}$ and $\nabla^2\mathcal{L}(\boldsymbol{\theta}) = \widehat{\Gamma}$ where $(\widehat{\Gamma}, \widehat{\gamma}) = \Big(\frac{X^\top X}{n}, \frac{X^\top \boldsymbol{y}}{n}\Big)$, we need to show below that

$$\begin{aligned}\widehat{\boldsymbol{z}}_{U^c} &\le \frac{1}{\lambda_n}\left\{-\widehat{\Gamma}_{U^cU}\boldsymbol{\theta}^*_U + \widehat{\gamma}_{U^c} + \widehat{\Gamma}_{U^cU}\boldsymbol{\theta}^*_U - \widehat{\Gamma}_{U^cU}\Big(\widehat{\Gamma}_{UU}\Big)^{-1}\widehat{\gamma}_U\right\} + \left|\!\left|\!\left|\widehat{\Gamma}_{U^cU}\Big(\widehat{\Gamma}_{UU}\Big)^{-1}\right|\!\right|\!\right|_\infty \\ &\le \frac{1}{\lambda_n}\left\{\widehat{\gamma}_{U^c} - \widehat{\Gamma}_{U^cU}\Big(\widehat{\Gamma}_{UU}\Big)^{-1}\widehat{\gamma}_U\right\} + \eta \end{aligned} \tag{7.28}$$

for the strict dual feasibility from (7.27). As derived in [231], we can write

$$\left\|\widehat{\gamma}_{U^c} - \widehat{\Gamma}_{U^cU}\Big(\widehat{\Gamma}_{UU}\Big)^{-1}\widehat{\gamma}_U\right\|_\infty = \left\|\frac{X^\top_{U^c}\Pi\boldsymbol{\epsilon}}{n}\right\|_\infty \tag{7.29}$$

where Π is an orthogonal project matrix on X_U: $I - X_U(X_U^\top X_U)^{-1}X_U^\top$.

For any j, we define u_j such that $e_j^\top \frac{X^\top_{U^c}\Pi\boldsymbol{\epsilon}}{n} := u_j^\top\boldsymbol{\epsilon}$. Then we have

$$\|u_j\|_2^2 = \left\|\frac{\Pi X_{U^c}e_j}{n}\right\|_2^2 \le \left\|\frac{X_{U^c}e_j}{n}\right\|_2^2 \le \frac{c_u}{n}. \tag{7.30}$$

Hence by the sub-Gaussian tail bounds followed by a union bound, we can conclude that

$$\left\|\widehat{\gamma}_{U^c} - \widehat{\Gamma}_{U^cU}\Big(\widehat{\Gamma}_{UU}\Big)^{-1}\widehat{\gamma}_U\right\|_\infty \le C\sqrt{\frac{\log p}{n}} \tag{7.31}$$

with probability at least $1 - c\exp(-c'\log p)$ for *all* selections of T. We can establish have strict dual feasibility for any selection of T w.h.p, provided $\lambda_n > \frac{C}{1-\eta}\sqrt{\frac{\log p}{n}}$, and now turn to ℓ_∞ bounds. From (7.6), we have

$$\left\|\widehat{\Gamma}_{UU}\Big(\widehat{\Gamma}_{UU}\boldsymbol{\theta}^*_U - \widehat{\gamma}_U\Big)\right\|_\infty = \left\|\left(\frac{X_U^\top X_U}{n}\right)^{-1}\left(\frac{X_U^\top\boldsymbol{\epsilon}}{n}\right)\right\|_\infty. \tag{7.32}$$

Then for $j \in U$, we define v such that $e_j^\top\left(\frac{X_U^\top X_U}{n}\right)^{-1}\left(\frac{X_U^\top\boldsymbol{\epsilon}}{n}\right) := v_j^\top\boldsymbol{\epsilon}$. Since for any selection of T, $\|v_j\|_2^2$ is bounded as follows:

$$\|v_j\|_2^2 = \frac{1}{n^2}\left\|X_U\left(\frac{X_U^\top X_U}{n}\right)^{-1}e_j\right\|_2^2 = \frac{1}{n}\left|e_j^\top\left(\frac{X_U^\top X_U}{n}\right)^{-1}e_j\right|_2^2 \le \frac{c_u}{n}. \tag{7.33}$$

Similarly by the sub-Gaussian tail bound and a union bound over j, we can obtain

$$\left\|\widehat{\Gamma}_{UU}\Big(\widehat{\Gamma}_{UU}\boldsymbol{\theta}^*_U - \widehat{\gamma}_U\Big)\right\|_\infty \le C\sqrt{\frac{\log p}{n}} \tag{7.34}$$

with probability at least $1 - c\exp(-c'\log p)$.

Proof of Corollary 5

As in the proof of Corollary 4, the proof procedure is quite similar to that of Corollary 4 of [231]. Deriving upper bounds on S in [231] can be seamlessly extendable to upper bounds on U for any selection of $T \subseteq \{1,2,\ldots,p\} \times \{1,2,\ldots,p\}$ s.t. $|T| = h$. mainly because the required upper bounds are related to entry-wise maximum on the true support S but entry-wise maximum in this case is uniformly upper bounded for all entries.

Specifically, it computes the upper bound of $\|\mathrm{vec}(\widehat{\Sigma}_S - \Sigma^*_S)\|_\infty$ from the fact that $\|\mathrm{vec}(\widehat{\Sigma} - \Sigma^*)\|_\infty \leq c\sqrt{\frac{\log p}{n}}$. This actually holds for any selection of T. Similarly, it computes the upper bound of $\max_{(j,k)\in S} |e_j^\top (\Sigma^*\Delta)^\ell \Sigma^* e_k|$ by Hölder's inequality and the definition of matrix induced norms: $|e_j^\top (\Sigma^*\Delta)^\ell \Sigma^* e_k| \leq \|e_j^\top (\Sigma^*\Delta)^{\ell-1}\|_1 \|\Delta\Sigma^* e_k\|_\infty \leq \left\|\!\left|(\Sigma^*\Delta)^{\ell-1}\right|\!\right\|_\infty \|\Delta\|_{\max} \|\Sigma^* e_k\|_1 \leq \|\!|\Sigma^*|\!\|_\infty^{\ell+1} \|\!|\Delta|\!\|_1^{\ell-1} \|\Delta\|_{\max}$, which clearly holds for any index (j,k) beyond S. Finally, $\|\!|\widehat{Q}_{SS} - \nabla^2\mathcal{L}(\mathbf{\Theta}^*)_{SS}|\!\|_\infty$ is shown to be upper bounded by the fact that $\|\!|\widehat{Q}_{SS} - \nabla^2\mathcal{L}(\mathbf{\Theta}^*)_{SS}|\!\|_\infty \precsim d\sqrt{\frac{\log p}{n}}$.

The remaining proof of this result directly follows similar lines to the proof of Corollary 4 in [231].

Proof of Theorem 19

Proof. From Algorithm 14, we obtain the relation

$$\begin{aligned}
\frac{1}{\tau}(\boldsymbol{w}^k - \boldsymbol{w}^{k+1}) + r(\boldsymbol{\theta}^{k+1}) - r(\boldsymbol{\theta}^k) &\in \boldsymbol{r}(\boldsymbol{\theta}^{k+1}) + \partial\delta(\boldsymbol{w}^{k+1}|\mathcal{S}) \\
\frac{1}{\eta}(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k+1}) + \nabla\mathcal{L}(\boldsymbol{\theta}^{k+1}) - \nabla\mathcal{L}(\boldsymbol{\theta}^k) &\in \nabla\mathcal{L}(\boldsymbol{\theta}^{k+1}) + \lambda\sum_{i=1}^p w_i^{k+1}\partial r_i(\boldsymbol{\theta}^{k+1})
\end{aligned}$$

from the proximal gradient steps.

At k-th iteration, we have

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}^{k+1}) + \lambda\langle \boldsymbol{w}^{k+1}, \boldsymbol{r}(\boldsymbol{\theta}^{k+1})\rangle \\
\leq&\mathcal{L}(\boldsymbol{\theta}^k) + \langle \nabla\mathcal{L}(\boldsymbol{\theta}^k), \boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\rangle + \frac{L}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \lambda\langle \boldsymbol{w}^{k+1}, \boldsymbol{r}(\boldsymbol{\theta}^{k+1})\rangle \\
\leq&\mathcal{L}(\boldsymbol{\theta}^k) + \langle \nabla\mathcal{L}(\boldsymbol{\theta}^k) + \lambda\sum_{i=1}^p w_i^{k+1}\partial r_i(\boldsymbol{\theta}^{k+1}), \boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\rangle + \frac{L}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \lambda\langle \boldsymbol{w}^{k+1}, \boldsymbol{r}(\boldsymbol{\theta}^k)\rangle \\
=&\mathcal{L}(\boldsymbol{\theta}^k) + \lambda\langle \boldsymbol{w}^k, \boldsymbol{r}(\boldsymbol{\theta}^k)\rangle - (\frac{1}{\eta} - \frac{L}{2})\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \lambda\langle \boldsymbol{w}^{k+1} - \boldsymbol{w}^k, \boldsymbol{r}(\boldsymbol{\theta}^k)\rangle \\
\leq&\mathcal{L}(\boldsymbol{\theta}^k) + \lambda\langle \boldsymbol{w}^k, \boldsymbol{r}(\boldsymbol{\theta}^k)\rangle - (\frac{1}{\eta} - \frac{L}{2})\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \lambda\langle \boldsymbol{w}^{k+1} - \boldsymbol{w}^k, \frac{1}{\tau}(\boldsymbol{w}^k - \boldsymbol{w}^{k+1}) - \partial\delta(\boldsymbol{w}^{k+1})\rangle \\
\leq&f(\boldsymbol{\theta}^k) + \lambda\langle \boldsymbol{w}^k, \boldsymbol{r}(\boldsymbol{\theta}^k)\rangle - (\frac{1}{\eta} - \frac{L}{2})\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 - \frac{\lambda}{\tau}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2
\end{aligned}$$

If we choose $\eta = 1/L_f$, we have,

$$\frac{L_f}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \frac{\lambda}{\tau}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2 \leq F(\boldsymbol{\theta}^k) - F(\boldsymbol{\theta}^{k+1})$$

By telescoping both sides we get,

$$\frac{1}{K}\sum_{k=1}^{K}(\frac{L_f}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|^2 + \frac{\lambda}{\tau}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2) \leq \frac{1}{K}(F(\boldsymbol{\theta}^K) - F^*).$$

Moreover we know that,

$$T(\boldsymbol{\theta}^{k+1}, \boldsymbol{w}^{k+1}) \leq (4 + 2\lambda L_r/L_f)\mathcal{G}_k.$$

□

7.6.3 Simulations for Gaussian Graphical Models.

We now illustrate the usefulness trimmed regularization for sparse Gaussian Graphical Model estimation. We consider the "diamond" graph example described in [287] (section 3.1.1). This graph $G = (V, E)$ has vertex set $V = \{1, 2, 3, 4\}$, with all edges except $(1, 4)$. We consider a family of true covariance matrices with diagonal entries $\Sigma^*_{ii} = 1$ for all $i \in V$; off-diagonal elements $\Sigma^*_{ij} = \rho$ for all edges $(i, j) \in E \setminus \{(2, 3)\}$; $\Sigma^*_{23} = 0$; and finally the entry corresponding to the non-edge $(1, 4)$ is set as $\Sigma^*_{14} = 2\rho^2$. We analyze the performance of Graphical Trimmed Lasso under two settings: $\rho \in \{0, 1, 0.3\}$. As discussed in [287], if $\rho = 0.1$ the incoherence condition is satisfied ; if $\rho = 0.3$ it is violated. Under both settings, we report the probability of successful support recovery based on 100 replicate experiments for $n = 100$ and $\frac{p^2-h}{p^2} \in \{0.4, 0.5, ..., 1\}$ and compare it with Graphical Lasso, Graphical SCAD and Graphical MCP (The MCP and SCAD parameters were set to 2.5 and 3.0 as varying these did not affect the results significantly). For each method and replicate experiment, success is declared if the true support is recovered for at least one value of λ_n along the solution path. We can see that for a wide range of values for the trimming parameter, Graphical Trimmed Lasso outperforms SCAD and MCP alternatives regardless of whether the incoherence condition holds or not. In addition its probability of success is always superior to that of vanilla Graphical Lasso, which fails to recover the true support when the incoherence condition is violated.

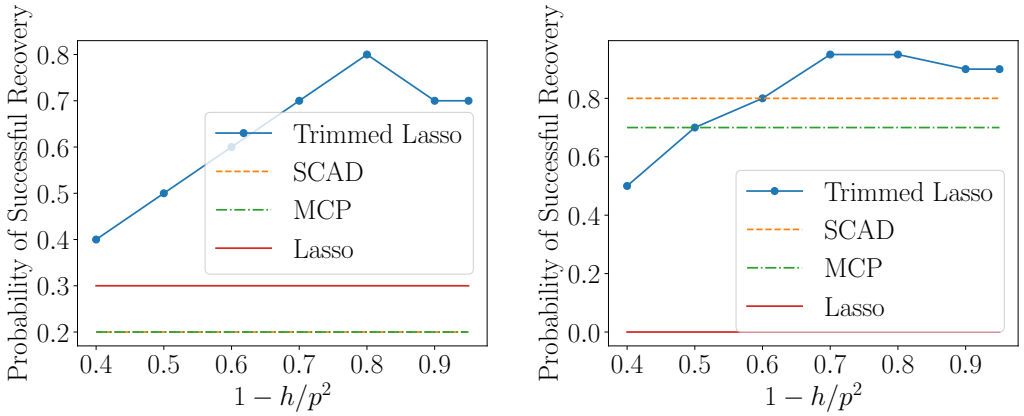


Figure 7.8: Probability of successful support recovery for Graphical Trimmed Lasso as h vary, Graphical SCAD, Graphical MCP and Graphical Lasso. Left: incoherence condition holds. Right: incoherence condition is violated.

BIBLIOGRAPHY

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Felipe Acker and Marc-Antoine Prestel. Convergence d'un schéma de minimisation alternée. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 2, pages 1–9. Université Paul Sabatier, 1980.

[3] A. Agarwal, S. Negahban, and M. J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *Annals of Statistics*, 40(2):1171–1197, 2012.

[4] F Al-Khayyal and J Kyparisis. Finite convergence of algorithms for nonlinear programs and variational inequalities. *Journal of Optimization Theory and Applications*, 70(2):319–332, 1991.

[5] Andreas Alfons, Christophe Croux, Sarah Gelper, et al. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013.

[6] BE Allman, PJ McMahon, KA Nugent, D Paganin, David L Jacobson, Muhammad Arif, and SA Werner. Imaging: phase radiography with neutrons. *Nature*, 408(6809):158, 2000.

[7] Massih-Reza Amini and Patrick Gallinari. Semi-supervised logistic regression. In *ECAI*, pages 390–394, 2002.

[8] A. Y. Aravkin and T. van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012.

[9] Aleksandr Aravkin and Stephen Becker. Dual smoothing and value function techniques for variational matrix decomposition. *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, page 2, 2016.

[10] Aleksandr Aravkin and Stephen Becker. Dual smoothing and value function techniques for variational matrix decomposition. In Thierry Bouwmans, Necdet Serhat

Aybat, and El-hadi Zahzah, editors, *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, chapter 3. CRC Press, 2016.

[11] Aleksandr Aravkin, James V Burke, Lennart Ljung, Aurelie Lozano, and Gianluigi Pillonetto. Generalized kalman smoothing: Modeling and algorithms. *Automatica*, 86:63–86, 2017.

[12] Aleksandr Aravkin and Damek Davis. A smart stochastic algorithm for nonconvex optimization with applications to robust machine learning. *arXiv preprint arXiv:1610.01101*, 2016.

[13] Aleksandr Aravkin, Dmitriy Drusvyatskiy, and Tristan van Leeuwen. Variable projection without smoothness. *arXiv preprint arXiv:1601.05011*, 2016.

[14] Aleksandr Aravkin, Rajiv Kumar, Hassan Mansour, Ben Recht, and Felix J Herrmann. Fast methods for denoising matrix completion formulations, with applications to robust seismic data interpolation. *SIAM Journal on Scientific Computing*, 36(5):S237–S266, 2014.

[15] Aleksandr Y Aravkin, Dmitriy Drusvyatskiy, and Tristan van Leeuwen. Efficient quadratic penalization through the partial minimization technique. *IEEE Transactions on Automatic Control*, 2017.

[16] Aleksandr Y Aravkin, Dmitriy Drusvyatskiy, and Tristan van Leeuwen. Efficient quadratic penalization through the partial minimization technique. *IEEE Transactions on Automatic Control*, 63(7):2131–2138, 2018.

[17] Aleksandr Y Aravkin and Tristan Van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012.

[18] Aleksandr Y Aravkin and Tristan Van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012.

[19] Travis Askham. duqbo/optdmd: optdmd v1.0.0, March 2017.

[20] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *arXiv preprint arXiv:1704.02343*, 2017.

[21] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[22] Travis Askham, Peng Zheng, Aleksandr Y Aravkin, and Nathan J Kutz. Robust and scalable methods for the dynamic mode decomposition. *arXiv preprint arXiv:1712.01883*, 2017.

[23] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[24] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[25] A Babiano, G Boffetta, A Provenzale, and A Vulpiani. Chaotic advection in point vortex models and two-dimensional turbulence. *Physics of Fluids*, 6(7):2465–2474, 1994.

[26] F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9:1019–1048, June 2008.

[27] F. Bach. Self-concordant analysis for logistic regression. *Electron. J. Stat.*, 4:384–414, 2010.

[28] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

[29] O. Bannerjee, , L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Jour. Mach. Lear. Res.*, 9:485–516, March 2008.

[30] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. Technical report, Rice University, 2008. Available at arxiv:0808.3572.

[31] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.

[32] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

[33] Heinz H Bauschke, Patrick L Combettes, and Dominikus Noll. Joint minimization with alternating bregman proximity operators. *Pacific Journal of Optimization*, 2(3):401–424, 2006.

[34] Edip Baysal, Dan D Kosloff, and John WC Sherwood. Reverse time migration. *Geophysics*, 48(11):1514–1524, 1983.

[35] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[36] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[37] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[38] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[39] Bradley M Bell and James V Burke. Algorithmic differentiation of implicit functions and optimal values. In *Advances in Automatic Differentiation*, pages 67–77. Springer, 2008.

[40] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[41] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[42] Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[43] Dimitris Bertsimas, Martin S Copenhaver, and Rahul Mazumder. The trimmed lasso: Sparsity and robustness. *arXiv preprint arXiv:1708.04527*, 2017.

[44] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*, 2012.

[45] P. J. Bickel and E. Levina. Covariance regularization by thresholding. *Annals of Statistics*, 36(6):2577–2604, 2008.

[46] P. J. Bickel and E. Levina. Covariance regularization by thresholding. *Annals of Statistics*, 36:2577–2604, 2008.

[47] P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Annals of Statistics*, 36:199–227, 2008.

[48] P. J. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

[49] P. J. Bickel, RitovY., and Tsybakov A. B. Simultaneous analysis of lasso and dantzig selector. *ANNALS OF STATISTICS*, 37(4), 2009.

[50] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[51] Małgorzata Bogdan, Ewout Van Den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J Candès. Slope?adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103, 2015.

[52] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[53] Thierry Bouwmans, Andrews Sobral, Sajid Javed, Soon Ki Jung, and El-Hadi Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23:1–71, 2017.

[54] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1—-122, 2011.

[55] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[56] Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics*, 5(1):232, 2011.

[57] Rasmus Bro. PARAFAC. Tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171, 1997.

[58] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

[59] Bingni W Brunton, Lise A Johnson, Jeffrey G Ojemann, and J Nathan Kutz. Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

[60] S. L. Brunton and B. R. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67:050801–1–050801–48, 2015.

[61] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[62] F. Bunea. Honest variable selection in linear and logistic regression models via l1 and l1 + l2 penalization. *Electron. J. Stat.*, 2:1153–1194, 2008.

[63] Samuel Burer and Renato D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Math. Program.*, 103(3):427–444, July 2005.

[64] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[65] J. V. Burke and M. C. Ferris. A gauss—newton method for convex composite optimization. *Mathematical Programming*, 71(2):179–194, 1995.

[66] James V Burke. Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3):260–279, 1985.

[67] James V Burke. An exact penalization viewpoint of constrained optimization. *SIAM Journal on control and optimization*, 29(4):968–998, 1991.

[68] James V Burke, Adrian S Lewis, and Michael L Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

[69] J.V. Burke. Descent methods for composite nondifferentiable optimization problems. *Math. Programming*, 33(3):260–279, 1985.

[70] JV Burke and Michael C Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993.

[71] T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

[72] T. Tony Cai, Zongming Ma, and Yihong Wu. Sparse PCA: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 12 2013.

[73] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.

[74] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *CPAM*, 59(8):1207–1223, 2006.

[75] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies. *Information Theory, IEEE Transactions on*, 52(12):5406 –5425, dec. 2006.

[76] E. J. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[77] E.J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925 –936, june 2010.

[78] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015.

[79] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[80] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[81] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[82] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[83] C. Cartis, N.I.M. Gould, and P.L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21(4):1721–1739, 2011.

[84] Stanley H Chan, Ramsin Khoshabeh, Kristofor B Gibson, Philip E Gill, and Truong Q Nguyen. An augmented lagrangian method for total variation video restoration. *IEEE Transactions on Image Processing*, 20(11):3097–3111, 2011.

[85] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998.

[86] Le Chang and Doris Y Tsao. The code for facial identity in the primate brain. *Cell*, 169(6):1013–1028, 2017.

[87] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233, 2008.

[88] Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011, 2011.

[89] Feishe Chen, Lixin Shen, and Bruce W Suter. Computing the proximity operator of the ℓ_p norm with $0 < p < 1$. *IET Signal Processing*, 10(5):557–565, 2016.

[90] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[91] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[92] Ward Cheney and Allen A Goldstein. Proximity maps for convex sets. *Proceedings of the American Mathematical Society*, 10(3):448–450, 1959.

[93] Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.

[94] Jon F Claerbout and Francis Muir. Robust modeling with erratic data. *Geophysics*, 38(5):826–844, 1973.

[95] Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, and Shen Chen Xu. Solving sdd linear systems in nearly mlog1/2n time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 343–352. ACM, 2014.

[96] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[97] Pierre Comon. Tensors: A brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53, 2014.

[98] Andrew R Conn. Constrained optimization using a nondifferentiable penalty function. *SIAM Journal on Numerical Analysis*, 10(4):760–784, 1973.

[99] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[100] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[101] Ludwig Cromme. Strong uniqueness. *Numerische Mathematik*, 29(2):179–193, 1978.

[102] Christophe Croux, Peter Filzmoser, and Heinrich Fritz. Robust sparse principal component analysis. *Technometrics*, 55(2):202–214, 2013.

[103] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16(1):2859–2900, 2015.

[104] Bin Dai, Chen Zhu, Baining Guo, and David Wipf. Compressing neural networks using the variational information bottleneck. In *International Conference on Machine learning (ICML)*, 2018.

[105] Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[106] Alexandre d'Aspremont, Laurent El Ghaoui, Michael I. Jordan, and Gert R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[107] Damek Davis and Dmitriy Drusvyatskiy. Stochastic subgradient method converges at the rate $O(k^{-1/4})$ on weakly convex functions. *arXiv preprint arXiv:1802.02988*, 2018.

[108] Damek Davis, Dmitriy Drusvyatskiy, and Courtney Paquette. The nonsmooth landscape of phase retrieval. *arXiv preprint arXiv:1711.03247*, 2017.

[109] Scott TM Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):1–19, 2016.

[110] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[111] Arnold Jan Den Dekker and A Van den Bos. Resolution: a survey. *JOSA A*, 14(3):547–557, 1997.

[112] Junjing Deng, David J Vine, Si Chen, Youssef SG Nashed, Qiaoling Jin, Nicholas W Phillips, Tom Peterka, Rob Ross, Stefan Vogt, and Chris J Jacobsen. Simultaneous cryo x-ray ptychographic and fluorescence microscopy of green algae. *Proceedings of the National Academy of Sciences*, 112(8):2314–2319, 2015.

[113] Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled clinical trials*, 7(3):177–188, 1986.

[114] Caglayan Dicle, Hassan Mansour, Dong Tian, Mouhacine Benosman, and Anthony Vetro. Robust low rank dynamic mode decomposition for compressed domain crowd and traffic flow analysis. In *Multimedia and Expo (ICME), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[115] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[116] David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4273–4293, 2009.

[117] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[118] AJJ Drenth, AMJ Huiser, and HA Ferwerda. The problem of phase retrieval in light and electron microscopy of strong objects. *Optica Acta: International Journal of Optics*, 22(7):615–628, 1975.

[119] Petros Drineas and Michael W Mahoney. RandNLA: Randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[120] D. Drusvyatskiy and A.S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *To appear in Math. Oper. Res., arXiv:1602.06661*, 2016.

[121] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Preprint arXiv:1605.00125*, 2016.

[122] Dmitriy Drusvyatskiy. The proximal point method revisited. *arXiv preprint arXiv:1712.06038*, 2017.

[123] Dmitriy Drusvyatskiy, Alexander D Ioffe, and Adrian S Lewis. Nonsmooth optimization using taylor-like models: error bounds, convergence, and termination criteria. *arXiv preprint arXiv:1610.03446*, 2016.

[124] Dmitriy Drusvyatskiy and C Kempton. Variational analysis of spectral functions simplified. *arXiv preprint arXiv:1506.05170*, 2015.

[125] J.C. Duchi and F. Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *Preprint arXiv:1705.02356*, 2017.

[126] J.C. Duchi and F. Ruan. Stochastic methods for composite optimization problems. *Preprint arXiv:1703.08570*, 2017.

[127] John C Duchi and Feng Ruan. Solving (most) of a set of quadratic equalities: Composite optimization for robust phase retrieval. *arXiv preprint arXiv:1705.02356*, 2017.

[128] Alexandre d'Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294, 2008.

[129] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[130] N. El Karoui. Operator norm consistent estimation of large dimensional sparse covariance matrices. *Annals of Statistics*, 36:2717–2756, 2008.

[131] N. El Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *Annals of Statistics*, 36:2757–2790, 2008.

[132] Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007.

[133] Yonina C Eldar and Shahar Mendelson. Phase retrieval: Stability and recovery guarantees. *Applied and Computational Harmonic Analysis*, 36(3):473–494, 2014.

[134] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[135] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Randomized dynamic mode decomposition. *arXiv preprint arXiv:1702.02912*, 2017.

[136] N. B. Erichson, K. Manohar, S. L. Brunton, and J. N. Kutz. Randomized CP tensor decomposition. *arXiv preprint arXiv:1703.09074*, 2017.

[137] N Benjamin Erichson, Steven L Brunton, and J Nathan Kutz. Compressed dynamic mode decomposition for real-time object detection. *Preprint*, 2015.

[138] N Benjamin Erichson, Sergey Voronin, Steven L Brunton, and J Nathan Kutz. Randomized matrix decompositions using R. *arXiv preprint arXiv:1608.02148*, 2016.

[139] Ruben Heras Evangelio, Michael Pätzold, and Thomas Sikora. Splitting gaussians in mixture models. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 300–305. IEEE, 2012.

[140] J. Fan and R. Li. Variable selection via non-concave penalized likelihood and its oracle properties. *Jour. Amer. Stat. Ass.*, 96(456):1348–1360, December 2001.

[141] J. Fan and J. Lv. Sure independence screening for ultra-high dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) (JRSSB)*, 70:849–911, 2008.

[142] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[143] Michael W Farn. New iterative algorithm for the design of phase-only gratings. In *Computer and Optically Generated Holographic Optics; 4th in a Series*, volume 1555, pages 34–43. International Society for Optics and Photonics, 1991.

[144] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[145] James R Fienup. Reconstruction of an object from the modulus of its fourier transform. *Optics letters*, 3(1):27–29, 1978.

[146] James R Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

[147] JR Fienup. Iterative method applied to image reconstruction and to computer-generated holograms. *Optical Engineering*, 19(3):193297, 1980.

[148] Mario AT Figueiredo and Robert D Nowak. Sparse estimation with strongly correlated variables using ordered weighted l1 regularization. *arXiv preprint arXiv:1409.4005*, 2014.

[149] R Fletcher. A model algorithm for composite nondifferentiable optimization problems. In *Nondifferential and Variational Techniques in Optimization*, pages 67–76. Springer, 1982.

[150] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[151] Joachim Frank, Pawel Penczek, Rajendra K Agrawal, Robert A Grassucci, and Amy B Heagle. [18] three-dimensional cryoelectron microscopy of ribosomes. 2000.

[152] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 2007.

[153] Gary Froyland and Michael Dellnitz. Detecting and locating near-optimal almost-invariant sets and cycles. *SIAM Journal on Scientific Computing*, 24(6):1839–1863, 2003.

[154] S. Funk. Netflix update: Try this at home, December 2006.

[155] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[156] Hong-Ye Gao and Andrew G Bruce. Waveshrink with firm shrinkage. *Statistica Sinica*, pages 855–874, 1997.

[157] Carl Friedrich Gauss. Theoria motus corporum coelestum. *Werke*, 1809.

[158] CF Gauss. Theory of the combination of observations which leads to the smallest errors. *Gauss Werke*, 4:1–93, 1821.

[159] C. R. Genovese, J. Jin, L. Wasserman, and Z. Yao. A comparison of the lasso and marginal regression. *Journal of Machine Learning Research (JMLR)*, 13:2107–2143, 2012.

[160] Ralph W Gerchberg. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

[161] N. Gillis. Introduction to nonnegative matrix factorization. *SIAG/OPT Views and News*, 25(1):7–16, 2017.

[162] Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[163] G. H. Golub and R. J. LeVeque. Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems. In *Proceedings of the 1979 Army Numerical Analsysis and Computers Conference*, 1979.

[164] Gene Golub and Victor Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse problems*, 19(2):R1, 2003.

[165] Gene Golub and Victor Pereyra. Separable nonlinear least squares: The variable projection method and its applications. *Inverse Problems*, 19(2):R1–R26, 2003.

[166] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[167] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[168] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[169] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[170] Jun-ya Gotoh, Akiko Takeda, and Katsuya Tono. Dc formulations and algorithms for sparse optimization problems. *Mathematical Programming*, pages 1–36, 2017.

[171] John C Gower and Garmt B Dijksterhuis. *Procrustes problems*, volume 30. Oxford University Press, 2004.

[172] J. Grosek and J. N. Kutz. *Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video. arXiv preprint, arXiv:1404.7592*, 2014.

[173] David Gross. Recovering Low-Rank Matrices From Few Coefficients in Any Basis. *IEEE Transactions on Information Theory*, 57:1548–1566, 2011.

[174] Yue Guan and Jennifer G Dy. Sparse probabilistic principal component analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 185–192, 2009.

[175] Florimond Guéniat, Lionel Mathelin, and Luc R. Pastur. A dynamic mode decomposition approach for large and arbitrarily sampled systems. *Physics of Fluids*, 27(2):025113, 2015.

[176] Charles Guyon, Thierry Bouwmans, and El-hadi Zahzah. Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis. In *Principal component analysis*. InTech, 2012.

[177] Tom SF Haines and Tao Xiang. Background subtraction with dirichlet processes. In *European Conference on Computer Vision*, pages 99–113. Springer, 2012.

[178] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, May 2011.

[179] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[180] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[181] Warren Hare and Claudia Sagastizábal. Computing proximal points of nonconvex functions. *Mathematical Programming*, 116(1):221–258, Jan 2009.

[182] Robert W Harrison. Phase problem in crystallography. *JOSA a*, 10(5):1046–1055, 1993.

[183] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[184] R. He, W. Zheng, T. Tan, and Z. Sun. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):261–275, Feb 2014.

[185] Maziar S Hemati, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Debiasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets. *Theoretical and Computational Fluid Dynamics*, pages 1–20, 2017.

[186] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[187] Rainer Hettich. A review of numerical methods for semi-infinite optimization. In *Semi-infinite programming and applications*, pages 158–178. Springer, 1983.

[188] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*, page 1, 2011.

[189] Arthur E Hoerl and Robert W Kennard. Ridge regression iterative estimation of the biasing parameter. *Communications in Statistics-Theory and Methods*, 5(1):77–88, 1976.

[190] G. E. Hoffman, B. A. Logsdon, and J. G. Mezey. Puma: A unified framework for penalized multiple regression analysis of gwas data. *Plos computational Biology*, 2013.

[191] Je Hyeong Hong, Christopher Zach, and Andrew Fitzgibbon. Revisiting the variable projection method for separable nonlinear least squares problems. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5947. IEEE, 2017.

[192] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[193] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research (JMLR)*, 12:3371–3412, 2011.

[194] Jianwen Huang, Jianjun Wang, Feng Zhang, and Wendong Wang. New sufficient conditions of signal recovery with tight frames via l1-analysis approach. *IEEE Access*, 2018.

[195] Peter J Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.

[196] Mia Hubert, Tom Reynkens, Eric Schmitt, and Tim Verdonck. Sparse PCA for high-dimensional data with outliers. *Technometrics*, 58(4):424–434, 2016.

[197] L. Jacob, G. Obozinski, and J. P. Vert. Group Lasso with Overlap and Graph Lasso. In *International Conference on Machine Learning (ICML)*, pages 433–440, 2009.

[198] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[199] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[200] Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, 12(3):531–547, 2003.

[201] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

[202] Eurika Kaiser, Bernd R Noack, Laurent Cordier, Andreas Spohn, Marc Segond, Markus Abel, Guillaume Daviller, Jan Östh, Siniša Krajnović, and Robert K Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.

[203] S. M. Kakade, O. Shamir, K. Sridharan, and A. Tewari. Learning exponential families in high-dimensions: Strong convexity and sparsity. In *Inter. Conf. on AI and Statistics (AISTATS)*, 2010.

[204] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[205] Koulik Khamaru and Martin J Wainwright. Convergence guarantees for a class of non-convex and non-smooth optimization problems. *arXiv preprint arXiv:1804.09629*, 2018.

[206] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[207] Durk P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representation (ICLR)*, 2014.

[208] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[209] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[210] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.

[211] J Nathan Kutz, Xing Fu, and Steven L Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

[212] Ming-Jun Lai, Yangyang Xu, and Wotao Yin. Improved iteratively reweighted least squares for unconstrained smoothed \ell_q minimization. *SIAM Journal on Numerical Analysis*, 51(2):927–957, 2013.

[213] Nan M Laird, James H Ware, et al. Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974, 1982.

[214] Kenneth L Lange, Roderick JA Little, and Jeremy MG Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

[215] S.L. Lauritzen. *Graphical models*. Oxford University Press, USA, 1996.

[216] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[217] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[218] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365–411, 2004.

[219] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence*, 27(5):684–698, 2005.

[220] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

[221] Naomi Ehrich Leonard, Derek A Paley, Francois Lekien, Rodolphe Sepulchre, David M Fratantoni, and Russ E Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[222] Adrian S Lewis. Nonsmooth analysis of eigenvalues. *Mathematical Programming*, 84(1):1–24, 1999.

[223] Adrian S Lewis and Michael L Overton. Nonsmooth optimization via bfgs. *Submitted to SIAM J. Optimiz*, pages 1–35, 2009.

[224] A.S. Lewis and S.J. Wright. A proximal method for composite minimization. *Math. Program.*, pages 1–46, 2015.

[225] L. Li and K. C. Toh. An inexact interior point method for l1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2:291–315, 2010.

[226] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. *Just relax and come clustering!: A convexification of k-means clustering*. Linköping University Electronic Press, 2011.

[227] Rong Liu and Hao Zhang. Segmentation of 3d meshes through spectral clustering. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 298–305. IEEE, 2004.

[228] N Locantore, JS Marron, DG Simpson, N Tripoli, JT Zhang, KL Cohen, Graciela Boente, Ricardo Fraiman, Babette Brumback, Christophe Croux, et al. Robust principal component analysis for functional data. *Test*, 8(1):1–73, 1999.

[229] P. Loh and M. J. Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[230] P. Loh and M. J. Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research (JMLR)*, 16:559–616, 2015.

[231] P. Loh and M. J. Wainwright. Support recovery without incoherence: A case for nonconvex regularization. *Annals of Statistics*, 45(6):2455–2482, 2017.

[232] Jean-Christophe Loiseau and Steven L Brunton. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.

[233] Christos Louizos, Max Welling, and Durk P Kingma. Learning sparse neural networks through ℓ_0 regularization. In *International Conference on Learning Representation (ICLR)*, 2018.

[234] K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. Technical Report arXiv:0903.1468, ETH Zurich, March 2009.

[235] A. C. Lozano, G. Swirszcz, and N. Abe. Group orthogonal matching pursuit for variable selection and prediction. In *Neur. Info. Proc. Sys (NIPS)*, 2009.

[236] D Russell Luke, James V Burke, and Richard G Lyon. Optical wavefront reconstruction: Theory and numerical methods. *SIAM review*, 44(2):169–224, 2002.

[237] Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

[238] Omid Madani, David M Pennock, and Gary W Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *Advances in neural information processing systems*, pages 873–880, 2005.

[239] Michael W Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[240] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, Jan. 2008.

[241] Andrew J Majda and John Harlim. *Filtering complex turbulent systems*. Cambridge University Press, 2012.

[242] Krithika Manohar, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. Data-driven sparse sensor placement for reconstruction. *IEEE Control Systems Magazine*, 38(3):63–86, 2018.

[243] Krithika Manohar, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Optimized sampling for multiscale dynamics. *arXiv preprint arXiv:1712.05085*, 2017.

[244] Stefano Marchesini, Yu-Chao Tu, and Hau-tieng Wu. Alternating projection, ptychographic imaging and phase synchronization. *Applied and Computational Harmonic Analysis*, 41(3):815–851, 2016.

[245] RARD Maronna, R Douglas Martin, and Victor Yohai. *Robust statistics*, volume 1. John Wiley & Sons, Chichester. ISBN, 2006.

[246] MATLAB Documentation. Sparse matrix operations.

[247] Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[248] Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

[249] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[250] N. Meinshausen and B. Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, 37(1):246–270, 2009.

[251] Deyu Meng, Qian Zhao, and Zongben Xu. Improve robustness of sparse PCA by l1-norm maximization. *Pattern Recognition*, 45(1):487 – 497, 2012.

[252] Jianwei Miao, Pambos Charalambous, Janos Kirz, and David Sayre. Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342, 1999.

[253] Rick P Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

[254] B.S. Mordukhovich. *Variational analysis and generalized differentiation. I*, volume 330 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2006. Basic theory.

[255] Aditya G Nair and Kunihiko Taira. Network-theoretic approach to sparsified discrete vortex dynamics. *Journal of Fluid Mechanics*, 768:549–571, 2015.

[256] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Arxiv preprint arXiv:1010.2731v1*, 2010.

[257] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[258] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[259] Yu. Nesterov. Modified Gauss-Newton scheme with worst case guarantees for global performance. *Optim. Methods Softw.*, 22(3):469–483, 2007.

[260] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[261] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[262] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[263] G. Obozinski, M. J. Wainwright, and M. I. Jordan. Union support recovery in high-dimensional multivariate regression. *Annals of Statistics*, 2010. To appear.

[264] Jung Hun Oh and Joseph O. Deasy. Inference of radio-responsive gene regulatory networks using the graphical lasso algorithm. *BMC Bioinformatics*, 15(S-7):S5, 2014.

[265] DW Oldenburg and RG Ellis. Efficient inversion of magnetotelluric data in two dimensions. *Physics of the earth and planetary interiors*, 81(1-4):177–200, 1993.

[266] V. Oropeza and M. Sacchi. Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. *Geophysics*, 76(3):V25–V32, 2011.

[267] MR Osborne. Separable least squares, variable projection, and the gauss-newton algorithm. *Electronic Transactions on Numerical Analysis*, 28(2):1–15, 2007.

[268] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[269] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In *Asian Conference on Computer Vision*, pages 361–377. Springer, 2016.

[270] Ricardo Otazo, Emmanuel Candès, and Daniel K Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 73(3):1125–1136, 2015.

[271] Samet Oymak. Learning compact neural networks with regularization. In *International Conference on Machine learning (ICML)*, 2018.

[272] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[273] Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, page 201318679, 2013.

[274] Christopher C Paige and Michael A Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8(1):43–71, 1982.

[275] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[276] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[277] Karl Pearson. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(2):559, 1901.

[278] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *IJCAI*, pages 1925–1931, 2016.

[279] Tomasz Pietrzykowski. An exact potential method for constrained maxima. *SIAM Journal on numerical analysis*, 6(2):299–304, 1969.

[280] BT Polyak. Sharp minima. institute of control sciences lecture notes, moscow, ussr, 1979. In *IIASA workshop on generalized Lagrangians and their applications, IIASA, Laxenburg, Austria*, 1979.

[281] M.J.D. Powell. General algorithms for discrete nonlinear approximation calculations. In *Approximation theory, IV (College Station, Tex., 1983)*, pages 187–218. Academic Press, New York, 1983.

[282] M.J.D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. *Math. Programming*, 29(3):297–303, 1984.

[283] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.

[284] Peter Radchenko and Gourab Mukherjee. Convex clustering via l1 fusion penalization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1527–1546, 2017.

[285] Ravi Ramamoorthi. Modeling illumination variation with spherical harmonics. *Face Processing: Advanced Modeling Methods*, pages 385–424, 2006.

[286] G. Raskutti, M. J. Wainwright, and B. Yu. Restricted eigenvalue properties for correlated gaussian designs. *Journal of Machine Learning Research (JMLR)*, 99:2241–2259, 2010.

[287] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

[288] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, Vol 52(3):471–501, 2010.

[289] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. arXiv 0706.4138, June 2007.

[290] B. Recht, M Fazel, and P.A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[291] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML '05 Proceedings of the 22nd international conference on Machine learning*, pages 713 – 719, 2005.

[292] Richard W Reynolds, Nick A Rayner, Thomas M Smith, Diane C Stokes, and Wanqiu Wang. An improved in situ and satellite sst analysis for climate. *Journal of climate*, 15(13):1609–1625, 2002.

[293] R. T. Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

[294] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[295] R.T. Rockafellar and R.J-B. Wets. *Variational Analysis*. Grundlehren der mathematischen Wissenschaften, Vol 317, Springer, Berlin, 1998.

[296] R.T. Rockafellar and R.J-B. Wets. *Variational Analysis*, volume 317. Springer, 1998.

[297] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[298] A. J. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association (Theory and Methods)*, 104:177–186, 2009.

[299] A. J. Rothman, E. Levina, and J. Zhu. A new approach to cholesky-based estimation of high-dimensional covariance matrices. *Biometrika*, 2010.

[300] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[301] Peter J Rousseeuw. Multivariate Estimation with High Breakdown Point. *Mathematical statistics and applications*, 8:283–297, 1985.

[302] C. W. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(3):997–1013, 2005.

[303] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[304] Hayden Schaeffer, Russel Caflisch, Cory D Hauck, and Stanley Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639, 2013.

[305] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, November 2008.

[306] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[307] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[308] Alexander Schrijver. On the history of the shortest path problem. *Doc. Math*, 155, 2012.

[309] Gregory Shakhnarovich and Baback Moghaddam. Face recognition in subspaces. In *Handbook of Face Recognition*, pages 19–49. Springer, 2011.

[310] Chen Shaobing and D Donoho. Basis pursuit. In *28th Asilomar conf. Signals, Systems Computers*, 1994.

[311] Haipeng Shen and Jianhua Z Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, 99(6):1015–1034, 2008.

[312] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[313] Christian D Sigg and Joachim M Buhmann. Expectation-maximization for sparse and non-negative PCA. In *Proceedings of the 25th international conference on Machine learning*, pages 960–967. ACM, 2008.

[314] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[315] Andrews Sobral, Thierry Bouwmans, and E-h Zahzah. Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos. *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing. CRC Press, Taylor and Francis Group*, 2016.

[316] Daniel A. Spielman and Shang-Hua Teng. Solving sparse, symmetric, diagonally-dominant linear systems in time $0(m^{1.31})$. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 416–427, 2003.

[317] Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Analysis Applications*, 35(3):835–885, 2014.

[318] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 246–252. IEEE, 1999.

[319] David Strong and Tony Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse problems*, 19(6):S165, 2003.

[320] Weijie Su, Małgorzata Bogdan, Emmanuel Candès, et al. False discoveries occur early on the lasso path. *The Annals of Statistics*, 45(5):2133–2150, 2017.

[321] Yoshihiko Susuki and Igor Mezic. Nonlinear Koopman modes and a precursor to power system swing instabilities. *IEEE Transactions on Power Systems*, 27(3):1182–1191, 2012.

[322] Abraham Szöke. Holographic microscopy with a complicated reference. *Journal of Imaging Science and Technology*, 41(4):332–341, 1997.

[323] Kunihiko Taira, Steven L Brunton, Scott Dawson, Clarence W Rowley, Tim Colonius, Beverley J McKeon, Oliver T Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55(12):4013–4041, 2017.

[324] Kunihiko Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.

[325] Kunihiko Taira, Aditya G. Nair, and Steven L. Brunton. Network structure of two-dimensional decaying isotropic turbulence. *Journal of Fluid Mechanics*, 795:R2, 2016.

[326] Naoya Takeishi, Yoshinobu Kawahara, Yasuo Tabei, and Takehisa Yairi. Bayesian dynamic mode decomposition. In *Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2814–2821, 2017.

[327] Jayaraman J Thiagarajan, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, and Bhavya Kailkhura. Robust local scaling using conditional quantiles of graph similarities. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*, pages 762–769. IEEE, 2016.

[328] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[329] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[330] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[331] AN Tihonov. Ob ustojchivosti obratnyh zadach. *On stability of inverse problems]. DAN SSSR–Reports of the USSR Academy of Sciences*, 39:195–198, 1943.

[332] Andrei Nikolajevits Tihonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math.*, 4:1035–1038, 1963.

[333] Vikrant Singh Tomar and Richard C Rose. Manifold regularized deep neural networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[334] Alessandro Treves. On the perceptual structure of face space. *BioSystems*, 40(1-2):189–196, 1997.

[335] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. *Signal Processing*, 86:572–602, April 2006. Special issue on "Sparse approximations in signal and image processing".

[336] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[337] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[338] S. van de Geer and P. Buhlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[339] S. van de Geer, P. Bühlmann, and S. Zhou. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 5:688–749, 2011.

[340] Ewout Van Den Berg and Michael P Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[341] V Vapnik and A Sterin. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 10(3):1495–1503, 1977.

[342] J. M. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3–5, 1975.

[343] Thomas Veit, Frédéric Cao, and Patrick Bouthemy. A maximality principle applied to a contrario motion detection. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–1061. IEEE, 2005.

[344] Dirk J Verschuur, AJ Berkhout, and CPA Wapenaar. Adaptive surface-related multiple elimination. *Geophysics*, 57(9):1166–1177, 1992.

[345] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[346] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 2009.

[347] John Von Neumann. *Functional Operators, Volume 2: The Geometry of Orthogonal Spaces*, volume 2. Princeton University Press, 1950.

[348] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Information Theory*, 55:2183–2202, May 2009.

[349] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Information Theory*, 55:2183–2202, May 2009.

[350] M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Info. Theory*, 55:2183–2202, 2009.

[351] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[352] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, pages 1–35, 2015.

[353] G.A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

[354] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.

[355] A. Wille, P. Zimmermann, and E. [and others] Vranova. Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biology*, 5, 2004.

[356] D. Witten and R. Tibshirani. Survival analysis with high-dimensional covariates. *Stat Methods Med Res.*, 19:29–51, 2010.

[357] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

[358] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[359] S.J. Wright. Convergence of an inexact algorithm for composite nonsmooth optimization. *IMA J. Numer. Anal.*, 10(3):299–321, 1990.

[360] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, pages 478–487, 2016.

[361] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012.

[362] Mengwei Xu, J Ye Jane, and Liwei Zhang. Smoothing augmented lagrangian method for nonsmooth constrained optimization problems. *Journal of Global Optimization*, 62(4):675–694, 2015.

[363] E. Yang and A. Lozano. Robust Gaussian Graphical Modeling with the Trimmed Graphical Lasso. In *Advances in Neural Information Processing Systems*, pages 2602–2610, 2015.

[364] E. Yang, A. Lozano, and A. Aravkin. High-Dimensional Trimmed Estimators: A General Framework for Robust Structured Estimation. *arXiv preprint arXiv:1605.08299*, 2016.

[365] E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research (JMLR)*, 16:3813–3847, 2015.

[366] Eunho Yang, Aurelie Lozano, and Aleksandr Aravkin. General family of trimmed estimators for robust high-dimensional data analysis. *Electronic Journal of Statistics*, 12:3519–3553, 2018.

[367] 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 Transactions on Pattern Analysis and Machine Intelligence*, 39(1):156–171, Jan 2017.

[368] Junfeng Yang and Yin Zhang. Alternating direction algorithms for \ell_1-problems in compressive sensing. *SIAM journal on scientific computing*, 33(1):250–278, 2011.

[369] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling*, 27:160–173, 2009.

[370] M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[371] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[372] Y. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Math. Programming*, 31(3):269–285, 1985.

[373] Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). In *Advances in neural information processing systems*, pages 1033–1040, 2002.

[374] Jihun Yun, Peng Zheng, Eunho Yang, Aurelie Lozano, and Aleksandr Y Aravkin. M-estimation with the trimmed l1 penalty. *ICML*, 2019.

[375] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2005.

[376] Cun-Hui Zhang et al. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010.

[377] Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, pages 576–593, 2012.

[378] J. Zhang, X. J. Jeng, and H. Liu. Some two-step procedures for variable selection in high-dimensional linear regression. *Arxiv preprint arXiv:0810.1644*, 2008.

[379] T. Zhang. Sparse recovery with orthogonal matching pursuit under rip. *IEEE Trans. Info. Theory*, 57:6215–6221, 2011.

[380] Tong Zhang. Multi-stage convex relaxation for learning with sparse regularization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1929–1936. Curran Associates, Inc., 2009.

[381] Tong Zhang. Sparse recovery with orthogonal matching pursuit under rip. *IEEE Trans. on Inform. Theory*, 57:6215 – 6221, 2011.

[382] P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2567, 2006.

[383] Peng Zheng and Aleksandr Aravkin. Relax-and-split method for nonsmooth nonconvex problems. *arXiv preprint arXiv:1802.02654*, 2018.

[384] Peng Zheng, Aleksandr Y Aravkin, Jayaraman J Thiagarajan, and Karthikeyan Natesan Ramamurthy. Learning robust representations for computer vision. In *ICCV Workshops*, pages 1784–1791, 2017.

[385] Peng Zheng, Travis Askham, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access*, 7:1404–1423, 2019.

[386] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[387] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[388] Alain Zuur, Elena N Ieno, Neil Walker, Anatoly A Saveliev, and Graham M Smith. *Mixed effects models and extensions in ecology with R*. Springer Science & Business Media, 2009.