Best subset selection via a modern optimization lens

Best Subset Selection via a Modern Optimization Lens

Dimitris Bertsimas* Angela King† Rahul Mazumder‡

(This is a Revised Version dated May, 2015. First Version Submitted for Publication on June, 2014.)

Abstract

In the last twenty-five years (1990-2014), algorithmic advances in integer optimization combined with hardware improvements have resulted in an astonishing 200 billion factor speedup in solving Mixed Integer Optimization (MIO) problems. We present a MIO approach for solving the classical best subset selection problem of choosing k out of p features in linear regression given n observations. We develop a discrete extension of modern first order continuous optimization methods to find high quality feasible solutions that we use as warm starts to a MIO solver that finds provably optimal solutions. The resulting algorithm (a) provides a solution with a guarantee on its suboptimality even if we terminate the algorithm early, (b) can accommodate side constraints on the coefficients of the linear regression and (c) extends to finding best subset solutions for the least absolute deviation loss function. Using a wide variety of synthetic and real datasets, we demonstrate that our approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds near optimal solutions for n in the 100s and p in the 1000s in minutes. We also establish via numerical experiments that the MIO approach performs better than `Lasso` and other popularly used sparse learning procedures, in terms of achieving sparse solutions with good predictive power.

*MIT Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology: dbertsim@mit.edu

†Operations Research Center, Massachusetts Institute of Technology: aking10@mit.edu

‡MIT Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology: rahulmaz@mit.edu

1 Introduction

We consider the linear regression model with response vector $\mathbf{y}_{n\times 1}$, model matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$, regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$ and errors $\boldsymbol{\epsilon} \in \mathbb{R}^{n\times 1}$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

We will assume that the columns of $\mathbf{X}$ have been standardized to have zero means and unit ℓ_2-norm. In many important classical and modern statistical applications, it is desirable to obtain a parsimonious fit to the data by finding the best k-feature fit to the response $\mathbf{y}$. Especially in the high-dimensional regime with $p \gg n$, in order to conduct statistically meaningful inference, it is desirable to assume that the true regression coefficient $\boldsymbol{\beta}$ is sparse or may be well approximated by a sparse vector. Quite naturally, the last few decades have seen a flurry of activity in estimating sparse linear models with good explanatory power. Central to this statistical task lies the best subset problem [40] with subset size k, which is given by the following optimization problem:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq k, \tag{1}$$

where the ℓ_0 (pseudo)norm of a vector $\boldsymbol{\beta}$ counts the number of nonzeros in $\boldsymbol{\beta}$ and is given by $\|\boldsymbol{\beta}\|_0 = \sum_{i=1}^{p} 1(\beta_i \neq 0)$, where $1(\cdot)$ denotes the indicator function. The cardinality constraint makes Problem (1) NP-hard [41]. Indeed, state-of-the-art algorithms to solve Problem (1), as implemented in popular statistical packages, like `leaps` in `R`, do not scale to problem sizes larger than $p = 30$. Due to this reason, it is not surprising that the best subset problem has been widely dismissed as being *intractable* by the greater statistical community.

In this paper we address Problem (1) using modern optimization methods, specifically mixed integer optimization (MIO) and a discrete extension of first order continuous optimization methods. Using a wide variety of synthetic and real datasets, we demonstrate that our approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds near optimal solutions for n in the 100s and p in the 1000s in minutes. To the best of our knowledge, this is the first time that MIO has been demonstrated to be a tractable solution method for Problem (1). We note that we use the term tractability not to mean the usual polynomial solvability for problems, but rather the ability to solve problems of realistic size in times that are appropriate for the applications we consider.

As there is a vast literature on the best subset problem, we next give a brief and selective overview of related approaches for the problem.

Brief Context and Background

To overcome the computational difficulties of the best subset problem, computationally tractable convex optimization based methods like `Lasso` [49, 17] have been proposed as a convex surrogate for Problem (1). For the linear regression problem, the Lagrangian form of `Lasso` solves

$$\min_{\boldsymbol{\beta}} \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2}$$

where the ℓ_1 penalty on $\boldsymbol{\beta}$, i.e., $\|\boldsymbol{\beta}\|_1 = \sum_i |\beta_i|$ shrinks the coefficients towards zero and naturally produces a sparse solution by setting many coefficients to be exactly zero. There has been a substantial amount of impressive work on `Lasso` [23, 15, 5, 55, 32, 59, 19, 35, 39, 53, 50] in terms of algorithms and understanding of its theoretical properties—see for example the excellent books or surveys [11, 34, 50] and the references therein.

Indeed, `Lasso` enjoys several attractive statistical properties and has drawn a significant amount of attention from the statistics community as well as other closely related fields. Under various conditions on the model matrix $\mathbf{X}$ and $n, p, \boldsymbol{\beta}$ it can be shown that `Lasso` delivers a sparse model with good predictive performance [11, 34]. In order to perform exact variable selection, much stronger assumptions are required [11]. Sufficient conditions under which `Lasso` gives a sparse model with good predictive performance are the restricted eigenvalue conditions and compatibility conditions [11]. These involve statements about the range of the spectrum of sub-matrices of $\mathbf{X}$ and are difficult to verify, for a given data-matrix $\mathbf{X}$.

An important reason behind the popularity of `Lasso` is its computational feasibility and scalability to practical sized problems. Problem (2) is a convex quadratic optimization problem and there are several efficient solvers for it, see for example [44, 23, 29].

In spite of its favorable statistical properties, `Lasso` has several shortcomings. In the presence of noise and correlated variables, in order to deliver a model with good predictive accuracy, `Lasso` brings in a large number of nonzero coefficients (all of which are shrunk towards zero) including noise variables. `Lasso` leads to biased regression coefficient estimates, since the ℓ_1-norm penalizes the large coefficients more severely than the smaller coefficients. In contrast, if the best subset selection procedure decides to include a variable in the model, it brings it in without any shrinkage thereby draining the effect of its correlated surrogates. Upon increasing the degree of regularization, `Lasso` sets more coefficients to zero, but in the process ends up leaving out true predictors from the active set. Thus, as soon as certain sufficient regularity conditions on the data are violated, `Lasso` becomes suboptimal as (a) a variable selector and (b) in terms of delivering a model with good predictive performance.

The shortcomings of `Lasso` are also known in the statistics literature. In fact, there

is a significant gap between what can be achieved via best subset selection and `Lasso`: this is supported by empirical (for small problem sizes, i.e., $p \leq 30$) and theoretical evidence, see for example, [46, 58, 38, 31, 56, 48] and the references therein. Some discussion is also presented herein, in Section 4.

To address the shortcomings, non-convex penalized regression is often used to "bridge" the gap between the convex ℓ_1 penalty and the combinatorial ℓ_0 penalty [38, 27, 24, 54, 55, 28, 61, 62, 57, 13]. Written in Lagrangian form, this gives rise to continuous non-convex optimization problems of the form:

$$\tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_i p(|\beta_i|; \gamma; \lambda), \tag{3}$$

where $p(|\beta|; \gamma; \lambda)$ is a non-convex function in β with λ and γ denoting the degree of regularization and non-convexity, respectively. Typical examples of non-convex penalties include the minimax concave penalty (MCP), the smoothly clipped absolute deviation (SCAD), and ℓ_γ penalties (see for example, [27, 38, 62, 24]). There is strong statistical evidence indicating the usefulness of estimators obtained as minimizers of non-convex penalized problems (3) over `Lasso` see for example [56, 36, 54, 25, 52, 37, 60, 26]. In a recent paper, [60] discuss the usefulness of non-convex penalties over convex penalties (like `Lasso`) in identifying important covariates, leading to efficient estimation strategies in high dimensions. They describe interesting connections between ℓ_0 regularized least squares and least squares with the hard thresholding penalty; and in the process develop comprehensive global properties of hard thresholding regularization in terms of various metrics. [26] establish asymptotic equivalence of a wide class of regularization methods in high dimensions with comprehensive sampling properties on both global and computable solutions.

Problem (3) mainly leads to a family of continuous and non-convex optimization problems. Various effective nonlinear optimization based methods (see for example [62, 24, 13, 36, 54, 38] and the references therein) have been proposed in the literature to obtain good local minimizers to Problem (3). In particular [38] proposes `Sparsenet`, a coordinate-descent procedure to trace out a surface of local minimizers for Problem (3) for the MCP penalty using effective warm start procedures. None of the existing approaches for solving Problem (3), however, come with guarantees of how close the solutions are to the global minimum of Problem (3).

The Lagrangian version of (1) given by

$$\tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=1}^{p} 1(\beta_i \neq 0), \tag{4}$$

may be seen as a special case of (3). Note that, due to non-convexity, problems (4) and (1) are *not* equivalent. Problem (1) allows one to control the exact level of sparsity via the choice of k, unlike (4) where there is no clear correspondence between λ

and k. Problem (4) is a discrete optimization problem unlike continuous optimization problems (3) arising from continuous non-convex penalties.

Insightful statistical properties of Problem (4) have been explored from a theoretical viewpoint in [56, 31, 32, 48]. [48] points out that (1) is preferable over (4) in terms of superior statistical properties of the resulting estimator. The aforementioned papers, however, do not discuss methods to obtain provably optimal solutions to problems (4) or (1), and to the best of our knowledge, computing optimal solutions to problems (4) and (1) is deemed as intractable.

Our Approach In this paper, we propose a novel framework via which the best subset selection problem can be solved to optimality or near optimality in problems of practical interest within a reasonable time frame. At the core of our proposal is a computationally tractable framework that brings to bear the power of modern discrete optimization methods: discrete first order methods motivated by first order methods in convex optimization [45] and mixed integer optimization (MIO), see [4]. We do not guarantee polynomial time solution times as these do not exist for the best subset problem unless P=NP. Rather, our view of computational tractability is the ability of a method to solve problems of practical interest in times that are appropriate for the application addressed. An advantage of our approach is that it adapts to variants of the best subset regression problem of the form:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta}} & \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_q^q \\ s.t. & \|\boldsymbol{\beta}\|_0 \leq k \\ & \mathbf{A}\boldsymbol{\beta} \leq \mathbf{b}, \end{array}$$

where $\mathbf{A}\boldsymbol{\beta} \leq \mathbf{b}$ represents polyhedral constraints and $q \in \{1, 2\}$ refers to a least absolute deviation or the least squares loss function on the residuals $\mathbf{r} := \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$.

Existing approaches in the Mathematical Optimization Literature In a seminal paper [30], the authors describe a leaps and bounds procedure for computing global solutions to Problem (1) (for the classical $n > p$ case) which can be achieved with computational effort significantly less than complete enumeration. `leaps`, a state-of-the-art `R` package uses this principle to perform best subset selection for problems with $n > p$ and $p \leq 30$. [3] proposed a tailored branch-and-bound scheme that can be applied to Problem (1) using ideas from [30] and techniques in quadratic optimization, extending and enhancing the proposal of [6]. The proposal of [3] concentrates on obtaining high quality upper bounds for Problem (1) and is less scalable than the methods presented in this paper.

Contributions We summarize our contributions in this paper below:

1. We use MIO to find a provably optimal solution for the best subset problem. Our approach has the appealing characteristic that if we terminate the algorithm early, we obtain a solution with a guarantee on its suboptimality. Furthermore, our framework can accommodate side constraints on $\boldsymbol{\beta}$ and also extends to finding best subset solutions for the least absolute deviation loss function.

2. We introduce a general algorithmic framework based on a discrete extension of modern first order continuous optimization methods that provide near-optimal solutions for the best subset problem. The MIO algorithm significantly benefits from solutions obtained by the first order methods and problem specific information that can be computed in a data-driven fashion.

3. We report computational results with both synthetic and real-world datasets that show that our proposed framework can deliver provably optimal solutions for problems of size n in the 1000s and p in the 100s in minutes. For high-dimensional problems with $n \in \{50, 100\}$ and $p \in \{1000, 2000\}$, with the aid of warm starts and further problem-specific information, our approach finds near optimal solutions in minutes but takes hours to prove optimality.

4. We investigate the statistical properties of best subset selection procedures for practical problem sizes, which to the best of our knowledge, have remained largely unexplored to date. We demonstrate the favorable predictive performance and sparsity-inducing properties of the best subset selection procedure over its competitors in a wide variety of real and synthetic examples for the least squares and absolute deviation loss functions.

The structure of the paper is as follows. In Section 2, we present a brief overview of MIO, including a summary of the computational advances it has enjoyed in the last twenty-five years. We present the proposed MIO formulations for the best subset problem as well as some connections with the compressed sensing literature for estimating parameters and providing lower bounds for the MIO formulations that improve their computational performance. In Section 3, we develop a discrete extension of first order methods in convex optimization to obtain near optimal solutions for the best subset problem and establish its convergence properties—the proposed algorithm and its properties may be of independent interest. Section 4 briefly reviews some of the statistical properties of the best-subset solution, highlighting the performance gaps in prediction error, over regular `Lasso`-type estimators. In Section 5, we perform a variety of computational tests on synthetic and real datasets to assess the algorithmic and statistical performances of our approach for the least squares loss function for both the classical overdetermined case $n > p$, and the high-dimensional case $p \gg n$. In Section 6, we report computational results for the least absolute deviation loss function. In Section 7, we include our

concluding remarks. Due to space limitations, some of the material has been relegated to the Appendix.

2 Mixed Integer Optimization Formulations

In this section, we present a brief overview of MIO, including the simply astonishing advances it has enjoyed in the last twenty-five years. We then present the proposed MIO formulations for the best subset problem as well as some connections with the compressed sensing literature for estimating parameters. We also present completely data driven methods to estimate parameters in the MIO formulations that improve their computational performance.

2.1 Brief Background on MIO

The general form of a Mixed Integer Quadratic Optimization (MIQO) problem is as follows:

$$\begin{aligned} \min \quad & \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{a} \\ s.t. \quad & \mathbf{A}\boldsymbol{\alpha} \leq \mathbf{b} \\ & \alpha_i \in \{0,1\}, \quad \forall i \in \mathcal{I} \\ & \alpha_j \in \mathbb{R}_+, \quad \forall j \notin \mathcal{I}, \end{aligned}$$

where $\mathbf{a} \in \mathbb{R}^m, \mathbf{A} \in \mathbb{R}^{k \times m}, \mathbf{b} \in \mathbb{R}^k$ and $\mathbf{Q} \in \mathbb{R}^{m \times m}$ (positive semidefinite) are the given parameters of the problem; $\mathbb{R}_+$ denotes the non-negative reals, the symbol $\leq$ denotes element-wise inequalities and we optimize over $\boldsymbol{\alpha} \in \mathbb{R}^m$ containing both discrete $(\alpha_i, i \in \mathcal{I})$ and continuous $(\alpha_i, i \notin \mathcal{I})$ variables, with $\mathcal{I} \subset \{1, \ldots, m\}$. For background on MIO see [4]. Subclasses of MIQO problems include convex quadratic optimization problems ($\mathcal{I} = \emptyset$), mixed integer ($\mathbf{Q} = \mathbf{0}_{m \times m}$) and linear optimization problems ($\mathcal{I} = \emptyset, \mathbf{Q} = \mathbf{0}_{m \times m}$). Modern integer optimization solvers such as Gurobi and Cplex are able to tackle MIQO problems.

In the last twenty-five years (1991-2014) the computational power of MIO solvers has increased at an astonishing rate. In [7], to measure the speedup of MIO solvers, the same set of MIO problems were tested on the same computers using twelve consecutive versions of Cplex and version-on-version speedups were reported. The versions tested ranged from Cplex 1.2, released in 1991 to Cplex 11, released in 2007. Each version released in these years produced a speed improvement on the previous version, leading to a total speedup factor of more than 29,000 between the first and last version tested (see [7], [42] for details). Gurobi 1.0, a MIO solver which was first released

in 2009, was measured to have similar performance to CPLEX 11. Version-on-version speed comparisons of successive GUROBI releases have shown a speedup factor of more than 20 between GUROBI 5.5, released in 2013, and GUROBI 1.0 ([7], [42]). The combined machine-independent speedup factor in MIO solvers between 1991 and 2013 is 580,000. This impressive speedup factor is due to incorporating both theoretical and practical advances into MIO solvers. Cutting plane theory, disjunctive programming for branching rules, improved heuristic methods, techniques for preprocessing MIOs, using linear optimization as a black box to be called by MIO solvers, and improved linear optimization methods have all contributed greatly to the speed improvements in MIO solvers [7].

In addition, the past twenty years have also brought dramatic improvements in hardware. Figure 1 shows the exponentially increasing speed of supercomputers over the past twenty years, measured in billion floating point operations per second [1]. The hardware speedup from 1993 to 2013 is approximately $10^{5.5} \sim 320,000$. When both hardware and software improvements are considered, the overall speedup is approximately 200 billion! Note that the speedup factors cited here refer to mixed integer linear optimization problems, not MIQO problems. The speedup factors for MIQO problems are similar. MIO solvers provide both feasible solutions as well as lower bounds to the optimal value. As the MIO solver progresses towards the optimal solution, the lower bounds improve and provide an increasingly better guarantee of suboptimality, which is especially useful if the MIO solver is stopped before reaching the global optimum. In contrast, heuristic methods do not provide such a certificate of suboptimality.

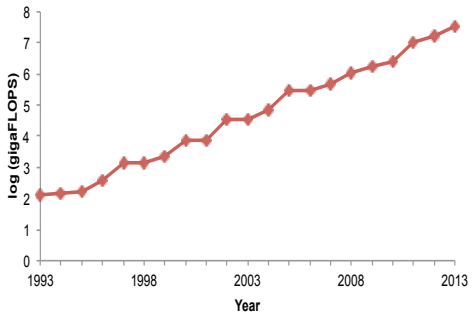

Figure 1: Log of Peak Supercomputer Speed from 1993–2013.

The belief that MIO approaches to problems in statistics are not practically relevant was formed in the 1970s and 1980s and it was at the time justified. Given the astonishing speedup of MIO solvers and computer hardware in the last twenty-five years, the mindset of MIO as theoretically elegant but practically irrelevant is no longer justified. In this paper, we provide empirical evidence of this fact in the context of the best subset selection problem.

2.2 MIO Formulations for the Best Subset Selection Problem

We first present a simple reformulation to Problem (1) as a MIO (in fact a MIQO) problem:

$$\begin{array}{lll} Z_1 = \min\limits_{\boldsymbol{\beta},\mathbf{z}} & \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 & \\ s.t. & -\mathcal{M}_U z_i \leq \beta_i \leq \mathcal{M}_U z_i, i=1,\ldots,p & \\ & z_i \in \{0,1\}, i=1,\ldots,p & \\ & \sum\limits_{i=1}^{p} z_i \leq k, & \end{array} \tag{5}$$

where $\mathbf{z} \in \{0,1\}^p$ is a binary variable and $\mathcal{M}_U$ is a constant such that if $\widehat{\boldsymbol{\beta}}$ is a minimizer of Problem (5), then $\mathcal{M}_U \geq \|\widehat{\boldsymbol{\beta}}\|_\infty$. If $z_i = 1$, then $|\beta_i| \leq \mathcal{M}_U$ and if $z_i = 0$, then $\beta_i = 0$. Thus, $\sum_{i=1}^p z_i$ is an indicator of the number of non-zeros in $\boldsymbol{\beta}$.

Provided that $\mathcal{M}_U$ is chosen to be sufficently large with $\mathcal{M}_U \geq \|\widehat{\boldsymbol{\beta}}\|_\infty$, a solution to Problem (5) will be a solution to Problem (1). Of course, $\mathcal{M}_U$ is not known a priori, and a small value of $\mathcal{M}_U$ may lead to a solution different from (1). The choice of $\mathcal{M}_U$ affects the strength of the formulation and is critical for obtaining good lower bounds in practice. In Section 2.3 we describe how to find appropriate values for $\mathcal{M}_U$. Note that there are other MIO formulations, presented herein (See Problem (8)) that do not rely on a-priori specifications of $\mathcal{M}_U$. However, we will stick to formulation (5) for the time being, since it provides some interesting connections to the `Lasso`.

Formulation (5) leads to interesting insights, especially via the structure of the convex hull of its constraints, as illustrated next:

$$\begin{aligned} & \text{Conv}\left(\left\{\boldsymbol{\beta} : |\beta_i| \leq \mathcal{M}_U z_i, z_i \in \{0,1\}, i=1,\ldots,p, \sum_{i=1}^{p} z_i \leq k\right\}\right) \\ = \;& \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_\infty \leq \mathcal{M}_U, \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_U k\} \subseteq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_U k\}. \end{aligned}$$

Thus, the minimum of Problem (5) is lower-bounded by the optimum objective value of both the following convex optimization problems:

$$Z_2 := \min_{\boldsymbol{\beta}} \; \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 \qquad \text{subject to} \;\; \|\boldsymbol{\beta}\|_\infty \leq \mathcal{M}_U, \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_U k \tag{6}$$

$$Z_3 := \min_{\boldsymbol{\beta}} \; \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 \qquad \text{subject to} \;\; \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_U k, \tag{7}$$

where (7) is the familiar `Lasso` in constrained form. This is a weaker relaxation than formulation (6), which in addition to the ℓ_1 constraint on $\boldsymbol{\beta}$, has box-constraints controlling the values of the β_i's. It is easy to see that the following ordering exists: $Z_3 \leq Z_2 \leq Z_1$, with the inequalities being strict in most instances.

In terms of approximating the optimal solution to Problem (5), the MIO solver begins by first solving a continuous relaxation of Problem (5). The `Lasso` formulation (7) is weaker than this root node relaxation. Additionally, MIO is typically able to significantly improve the quality of the root node solution as the MIO solver progresses toward the optimal solution.

To motivate the reader we provide an example of the evolution (see Figure 2) of the MIO formulation (8) for the Diabetes dataset [23], with $n = 350, p = 64$ (for further details on the dataset see Section 5).

Since formulation (5) is sensitive to the choice of $\mathcal{M}_U$, we consider an alternative MIO formulation based on Specially Ordered Sets [4] as described next.

Formulations via Specially Ordered Sets Any feasible solution to formulation (5) will have $(1 - z_i)\beta_i = 0$ for every $i \in \{1, \ldots, p\}$. This constraint can be modeled via integer optimization using Specially Ordered Sets of Type 1 [4] (SOS-1). In an SOS-1 constraint, at most one variable in the set can take a nonzero value, that is

$$(1 - z_i)\beta_i = 0 \quad \Longleftrightarrow \quad (\beta_i, 1 - z_i) : \text{SOS-1},$$

for every $i = 1, \ldots, p$. This leads to the following formulation of (1):

$$\begin{aligned} \min_{\boldsymbol{\beta}, \mathbf{z}} \quad & \tfrac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \\ s.t. \quad & (\beta_i, 1 - z_i) : \text{SOS-1}, \ \ i = 1, \ldots, p \\ & z_i \in \{0, 1\}, i = 1, \ldots, p \\ & \sum_{i=1}^{p} z_i \leq k. \end{aligned} \tag{8}$$

We note that Problem (8) can in principle be used to obtain global solutions to Problem (1) — Problem (8) unlike Problem (5) does not require any specification of the parameter $\mathcal{M}_U$.

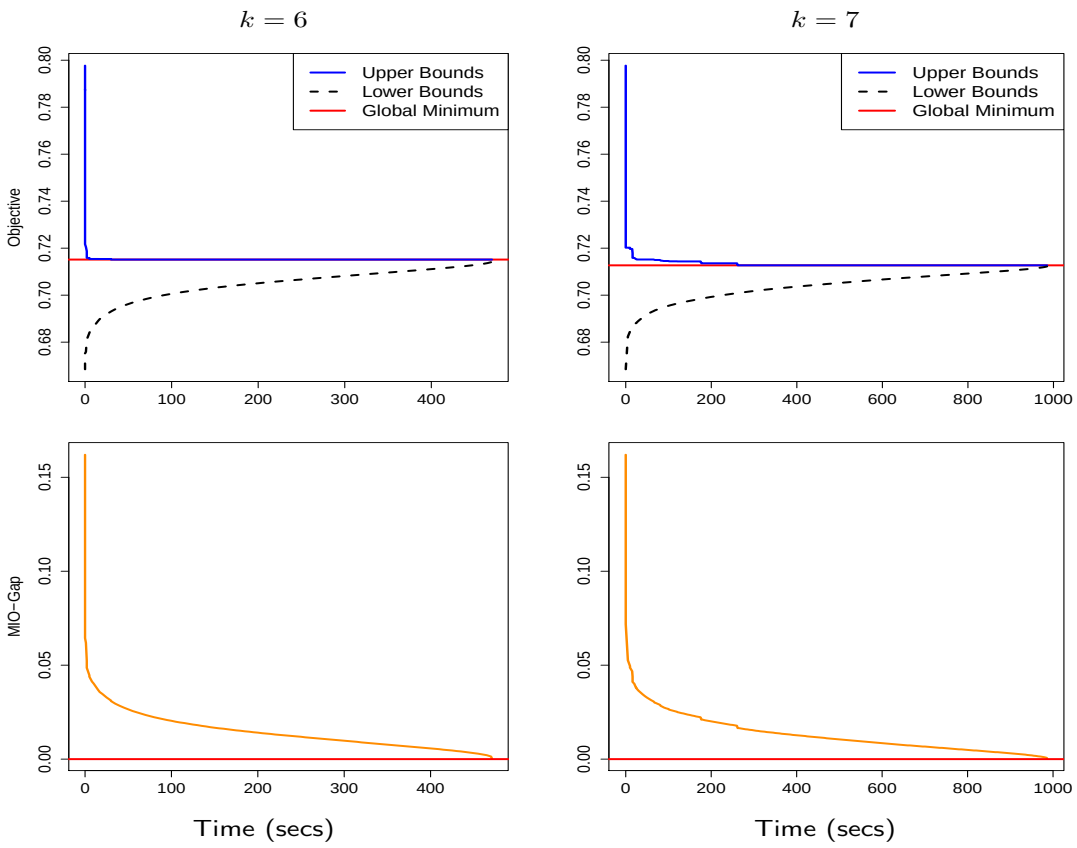

Figure 2: The typical evolution of the MIO formulation (8) for the diabetes dataset with $n = 350, p = 64$ with $k = 6$ (left panel) and $k = 7$ (right panel). The top panel shows the evolution of upper and lower bounds with time. The lower panel shows the evolution of the corresponding MIO gap with time. Optimal solutions for both the problems are found in a few seconds in both examples, but it takes 10-20 minutes to certify optimality via the lower bounds. Note that the time taken for the MIO to certify convergence to the global optimum increases with increasing k.

We now proceed to present a more structured representation of Problem (8). Note that objective in this problem is a convex quadratic function in the continuous variable $\boldsymbol{\beta}$,

which can be formulated explicitly as:

$$\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z}} \quad & \tfrac{1}{2}\,\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \tfrac{1}{2}\,\|\mathbf{y}\|_2^2 \\
s.t. \quad & (\beta_i, 1-z_i) : \text{SOS-1}, \;\; i = 1,\ldots,p \\
& z_i \in \{0,1\}, i = 1,\ldots,p \\
& \sum_{i=1}^{p} z_i \leq k \\
& -\mathcal{M}_U \leq \beta_i \leq \mathcal{M}_U, i = 1,\ldots,p \\
& \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell.
\end{aligned} \tag{9}$$

We also provide problem-dependent constants $\mathcal{M}_U$ and $\mathcal{M}_\ell \in [0,\infty]$. $\mathcal{M}_U$ provides an upper bound on the absolute value of the regression coefficients and $\mathcal{M}_\ell$ provides an upper bound on the ℓ_1-norm of $\boldsymbol{\beta}$. Adding these bounds typically leads to improved performance of the MIO, especially in delivering lower bound certificates. In Section 2.3, we describe several approaches to compute these parameters from the data.

We also consider another formulation for (9):

$$\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z},\boldsymbol{\zeta}} \quad & \tfrac{1}{2}\,\boldsymbol{\zeta}^T\boldsymbol{\zeta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \tfrac{1}{2}\,\|\mathbf{y}\|_2^2 \\
s.t. \quad & \boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta} \\
& (\beta_i, 1-z_i) : \text{SOS-1}, \;\; i = 1,\ldots,p \\
& z_i \in \{0,1\}, i = 1,\ldots,p \\
& \sum_{i=1}^{p} z_i \leq k \\
& -\mathcal{M}_U \leq \beta_i \leq \mathcal{M}_U, i = 1,\ldots,p \\
& \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell \\
& -\mathcal{M}_U^\zeta \leq \zeta_i \leq \mathcal{M}_U^\zeta, i = 1,\ldots,n \\
& \|\boldsymbol{\zeta}\|_1 \leq \mathcal{M}_\ell^\zeta,
\end{aligned} \tag{10}$$

where the optimization variables are $\boldsymbol{\beta} \in \mathbb{R}^p, \boldsymbol{\zeta} \in \mathbb{R}^n, \mathbf{z} \in \{0,1\}^p$ and $\mathcal{M}_U, \mathcal{M}_\ell, \mathcal{M}_U^\zeta, \mathcal{M}_\ell^\zeta \in [0,\infty]$ are problem specific parameters. Note that the objective function in formulation (10) involves a quadratic form in n variables and a linear function in p variables.

Problem (10) is equivalent to the following variant of the best subset problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta}} & \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 \\ s.t. & \|\boldsymbol{\beta}\|_0 \leq k \\ & \|\boldsymbol{\beta}\|_\infty \leq \mathcal{M}_U, \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell \\ & \|\mathbf{X}\boldsymbol{\beta}\|_\infty \leq \mathcal{M}_U^\zeta, \|\mathbf{X}\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell^\zeta. \end{array} \tag{11}$$

Formulations (9) and (10) differ in the size of the quadratic forms that are involved. The current state-of-the-art MIO solvers are better-equipped to handle mixed integer linear optimization problems than MIQO problems. Formulation (9) has fewer variables but a quadratic form in p variables—we find this formulation more useful in the $n > p$ regime, with p in the 100s. Formulation (10) on the other hand has more variables, but involves a quadratic form in n variables—this formulation is more useful for high-dimensional problems $p \gg n$, with n in the 100s and p in the 1000s.

As we said earlier, the bounds on $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ are not required, but if these constraints are provided, they improve the strength of the MIO formulation. In other words, formulations with tightly specified bounds provide better lower bounds to the global optimization problem in a specified amount of time, when compared to a MIO formulation with loose bound specifications. We next show how these bounds can be computed from given data.

2.3 Specification of Parameters

In this section, we obtain estimates for the quantities $\mathcal{M}_U, \mathcal{M}_\ell, \mathcal{M}_U^\zeta, \mathcal{M}_\ell^\zeta$ such that an optimal solution to Problem (11) is also an optimal solution to Problem (1), and vice-versa.

Coherence and Restricted Eigenvalues of a Model Matrix

Given a model matrix $\mathbf{X}$, [51] introduced the cumulative coherence function

$$\mu[k] := \max_{|I|=k} \max_{j \notin I} \sum_{i \in I} |\langle \mathbf{X}_j, \mathbf{X}_i \rangle|,$$

where, $\mathbf{X}_j$, $j = 1, \ldots, p$ represent the columns of $\mathbf{X}$, i.e., features.

For $k = 1$, we obtain the notion of coherence introduced in [22, 21] as a measure of the maximal pairwise correlation in absolute value of the columns of $\mathbf{X}$: $\mu := \mu[1] = \max_{i \neq j} |\langle \mathbf{X}_i, \mathbf{X}_j \rangle|$.

[16, 14] (see also [11] and references therein) introduced the notion that a matrix $\mathbf{X}$ satisfies a restricted eigenvalue condition if

$$\lambda_{\min}(\mathbf{X}_I'\mathbf{X}_I) \geq \eta_k \quad \text{for every } I \subset \{1,\ldots,p\} : \ |I| \leq k, \tag{12}$$

where $\lambda_{\min}(\mathbf{X}_I'\mathbf{X}_I)$ denotes the smallest eigenvalue of the matrix $\mathbf{X}_I'\mathbf{X}_I$. An inequality linking $\mu[k]$ and η_k is as follows.

Proposition 1. *The following bounds hold :*

(a) *[51]:* $\mu[k] \leq \mu \cdot k$.

(b) *[21] :* $\eta_k \geq 1 - \mu[k-1] \geq 1 - \mu \cdot (k-1)$.

The computations of $\mu[k]$ and η_k for general k are difficult, while μ is simple to compute. Proposition 1 provides bounds for $\mu[k]$ and η_k in terms of the coherence μ.

Operator Norms of Submatrices

The (p,q) operator norm of matrix $\mathbf{A}$ is

$$\|\mathbf{A}\|_{p,q} := \max_{\|\mathbf{u}\|_q=1} \|\mathbf{A}\mathbf{u}\|_p.$$

We will use extensively here the $(1,1)$ operator norm. We assume that each column vector of $\mathbf{X}$ has unit ℓ_2-norm. The results derived in the next proposition borrow and enhance techniques developed by [51] in the context of analyzing the ℓ_1—ℓ_0 equivalence in compressed sensing.

Proposition 2. *For any $I \subset \{1,\ldots,p\}$ with $|I| = k$ we have :*

(a) $\|\mathbf{X}_I'\mathbf{X}_I - \mathbf{I}\|_{1,1} \leq \mu[k-1]$.

(b) *If the matrix $\mathbf{X}_I'\mathbf{X}_I$ is invertible and $\|\mathbf{X}_I'\mathbf{X}_I - \mathbf{I}\|_{1,1} < 1$, then $\|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{1,1} \leq \frac{1}{1-\mu[k-1]}$.*

Proof. See Section A.3. □

We note that Part (b) also appears in [51] for the operator norm $\|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{\infty,\infty}$.

Given a set $I \subset \{1,\ldots,p\}$ with $|I| = k$ we let $\widehat{\boldsymbol{\beta}}_I$ denote the least squares regression coefficients obtained by regressing $\mathbf{y}$ on $\mathbf{X}_I$, i.e., $\widehat{\boldsymbol{\beta}}_I = (\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'\mathbf{y}$. If we append $\widehat{\boldsymbol{\beta}}_I$ with zeros in the remaining coordinates we obtain $\widehat{\boldsymbol{\beta}}$ as follows: $\widehat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}:\beta_i=0, i\notin I} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$. Note that $\widehat{\boldsymbol{\beta}}$ depends on I but we will suppress the dependence on I for notational convenience.

2.3.1 Specification of Parameters in terms of Coherence and Restricted Strong Convexity

Recall that $\mathbf{X}_j$, $j = 1, \ldots, p$ represent the columns of $\mathbf{X}$; and we will use $\mathbf{x}_i$, $i = 1, \ldots, n$ to denote the rows of $\mathbf{X}$. As discussed above $\|\mathbf{X}_j\| = 1$. We order the correlations $|\langle \mathbf{X}_j, \mathbf{y} \rangle|$:

$$|\langle \mathbf{X}_{(1)}, \mathbf{y} \rangle| \geq |\langle \mathbf{X}_{(2)}, \mathbf{y} \rangle| \ldots \geq |\langle \mathbf{X}_{(p)}, \mathbf{y} \rangle|. \tag{13}$$

We finally denote by $\|\mathbf{x}_i\|_{1:k}$ the sum of the top k absolute values of the entries of $x_{ij}, j \in \{1, 2, \ldots, p\}$.

Theorem 2.1. *For any $k \geq 1$ such that $\mu[k-1] < 1$ any optimal solution $\widehat{\boldsymbol{\beta}}$ to (1) satisfies:*

$$\textbf{(a)} \qquad \|\widehat{\boldsymbol{\beta}}\|_1 \leq \quad \frac{1}{1 - \mu[k-1]} \sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y} \rangle|. \tag{14}$$

$$\textbf{(b)} \qquad \|\widehat{\boldsymbol{\beta}}\|_\infty \leq \quad \min \left\{ \frac{1}{\eta_k} \sqrt{\sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y} \rangle|^2}, \frac{1}{\sqrt{\eta_k}} \|\mathbf{y}\|_2 \right\}. \tag{15}$$

$$\textbf{(c)} \qquad \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1 \leq \quad \min \left\{ \sum_{i=1}^{n} \|\mathbf{x}_i\|_\infty \|\widehat{\boldsymbol{\beta}}\|_1, \sqrt{k} \|\mathbf{y}\|_2 \right\}. \tag{16}$$

$$\textbf{(d)} \qquad \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty \leq \quad \left(\max_{i=1,\ldots,n} \|\mathbf{x}_i\|_{1:k} \right) \|\widehat{\boldsymbol{\beta}}\|_\infty. \tag{17}$$

Proof. For proof see Section A.4. □

We note that in the above theorem, the upper bound in Part (a) becomes infinite as soon as $\mu[k-1] \geq 1$. In such a case, we can use purely data-driven bounds by using convex optimization techniques, as described in Section 2.3.2.

The interesting message conveyed by Theorem 2.1 is that the upper bounds on $\|\widehat{\boldsymbol{\beta}}\|_1$, $\|\widehat{\boldsymbol{\beta}}\|_\infty$, $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1$ and $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty$, corresponding to the Problem (11) can all be obtained in terms of η_k and $\mu[k-1]$, quantities of fundamental interest appearing in the analysis of ℓ_1 regularization methods and understanding how close they are to ℓ_0 solutions [51, 22, 21, 16, 14]. On a different note, Theorem 2.1 arises from a purely computational motivation and quite curiously, involves the same quantities: cumulative coherence and restricted eigenvalues.

Note that the quantities $\mu[k-1], \eta_k$ are difficult to compute exactly, but they can be approximated by Proposition 1 which provides bounds commonly used in the compressed sensing literature. Of course, approximations to these quantities can also be obtained by using subsampling schemes.

2.3.2 Specification of Parameters via Convex Quadratic Optimization

We provide an alternative purely data-driven way to compute the upper bounds to the parameters by solving several simple convex quadratic optimization problems.

Bounds on $\hat{\beta}_i$'s

For the case $n > p$, upper and lower bounds on $\hat{\beta}_i$ can be obtained by solving the following pair of convex optimization problems:

$$\begin{array}{lll} u_i^+ := & \max\limits_{\boldsymbol{\beta}} & \beta_i \\ s.t. & & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \leq \text{UB}, \end{array} \qquad \begin{array}{lll} u_i^- := & \min\limits_{\boldsymbol{\beta}} & \beta_i \\ s.t. & & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \leq \text{UB}, \end{array} \tag{18}$$

for $i = 1, \ldots, p$. Above, UB is an upper bound to the minimum of the k-subset least squares problem (1). u_i^+ is an upper bound to $\hat{\beta}_i$, since the cardinality constraint $\|\boldsymbol{\beta}\|_0 \leq k$ does not appear in the optimization problem. Similarly, u_i^- is a lower bound to $\hat{\beta}_i$. The quantity $\mathcal{M}_U^i = \max\{|u_i^+|, |u_i^-|\}$ serves as an upper bound to $|\hat{\beta}_i|$. A reasonable choice for UB is obtained by using the discrete first order methods (Algorithms 1 and 2 as described in Section 3) in combination with the MIO formulation (8) (for a predefined amount of time). Having obtained $\mathcal{M}_U^i$ as described above, we can obtain an upper bound to $\|\widehat{\boldsymbol{\beta}}\|_\infty$ and $\|\widehat{\boldsymbol{\beta}}\|_1$ as follows: $\mathcal{M}_U = \max_i \mathcal{M}_U^i$ and $\|\widehat{\boldsymbol{\beta}}\|_1 \leq \sum_{i=1}^k \mathcal{M}_U^{(i)}$ where, $\mathcal{M}_U^{(1)} \geq \mathcal{M}_U^{(2)} \geq \ldots \geq \mathcal{M}_U^{(p)}$.

Similarly, bounds corresponding to Parts (c) and (d) in Theorem 2.1 can be obtained by using the upper bounds on $\|\widehat{\boldsymbol{\beta}}\|_\infty, \|\widehat{\boldsymbol{\beta}}\|_1$ as described above.

Note that the quantities u_i^+ and u_i^- are finite when the level sets of the least squares loss function are finite. In particular, the bounds are loose when $p > n$. In the following we describe methods to obtain non-trivial bounds on $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle$, for $i = 1, \ldots, n$ that apply for arbitrary n, p.

Bounds on $\langle \mathbf{x}_i, \widehat{\boldsymbol{\beta}} \rangle$'s

We now provide a generic method to obtain upper and lower bounds on the quantities $\langle \mathbf{x}_i, \widehat{\boldsymbol{\beta}} \rangle$:

$$\begin{array}{lll} v_i^+ := & \max\limits_{\boldsymbol{\beta}} & \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle \\ s.t. & & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \leq \text{UB}, \end{array} \qquad \begin{array}{lll} v_i^- := & \min\limits_{\boldsymbol{\beta}} & \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle \\ s.t. & & \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \leq \text{UB}, \end{array} \tag{19}$$

for $i = 1, \ldots, n$. Note that the bounds obtained from (19) are non-trivial bounds for both the under-determined $n < p$ and overdetermined cases. The bounds obtained from (19) are upper and lower bounds since we drop the cardinality constraint on $\boldsymbol{\beta}$. The bounds are finite since for every $i \in \{1, \ldots, n\}$ the quantity $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle$ remains bounded in the feasible set for Problems (19).

The quantity $\mathrm{v}_i = \max\{|v_i^+|, |v_i^-|\}$ serves as an upper bound to $|\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle|$. In particular, this leads to simple upper bounds on $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty \leq \max_i \mathrm{v}_i$ and $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1 \leq \sum_i \mathrm{v}_i$ and can be thought of completely data-driven methods to estimate bounds appearing in (16) and (17).

We note that Problems (18) and (19) have nice structure amenable to efficient computation as we discuss in Section A.1.

2.3.3 Parameter Specifications from Advanced Warm-Starts

The methods described above in Sections 2.3.1 and 2.3.2 lead to *provable* bounds on the parameters: with these bounds Problem (11) is provides an optimal solution to Problem (1), and vice-versa. We now describe some other alternatives that lead to excellent parameter specifications in practice.

The discrete first order methods described in the following section 3 provide good upper bounds to Problem (1). These solutions when supplied as a warm-start to the MIO formulation (8) are often improved by MIO, thereby leading to high quality solutions to Problem (1) within several minutes. If $\hat{\boldsymbol{\beta}}_{\text{hyb}}$ denotes an estimate obtained from this hybrid approach, then $\mathcal{M}_U := \tau \|\hat{\boldsymbol{\beta}}_{\text{hyb}}\|_\infty$ with τ a multiplier greater than one (e.g., $\tau \in \{1.1, 1.5, 2\}$) provides a good estimate for the parameter $\mathcal{M}_U$. A reasonable upper bound to $\|\widehat{\boldsymbol{\beta}}\|_1$ is $k\mathcal{M}_U$. Bounds on the other quantities: $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1, \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty$ can be derived by using expressions appearing in Theorem 2.1, with aforementioned bounds on $\|\widehat{\boldsymbol{\beta}}\|_1$ and $\|\widehat{\boldsymbol{\beta}}\|_\infty$.

2.3.4 Some Generalizations and Variants

Some variations and improvements of the procedures described above are presented in Section A.2 (appendix).

3 Discrete First Order Algorithms

In this section, we develop a discrete extension of first order methods in convex optimization [45, 44] to obtain near optimal solutions for Problem (1) and its variant for the least absolute deviation (LAD) loss function. Our approach applies to the problem of minimizing any smooth convex function subject to cardinality constraints.

We will use these discrete first order methods to obtain solutions to warm start the MIO formulation. In Section 5, we will demonstrate how these methods greatly enhance the performance of the MIO.

3.1 Finding stationary solutions for minimizing smooth convex functions with cardinality constraints

Related work and contributions In the signal processing literature [8, 9] proposed iterative hard-thresholding algorithms, in the context of ℓ_0-regularized least squares problems, i.e., Problem (4). The authors establish convergence properties of the algorithm under the assumption that $\mathbf{X}$ satisfies coherence [8] or Restricted Isometry Property [9]. The method we propose here applies to a larger class of cardinality constrained optimization problems of the form (20), in particular, in the context of Problem (1) our algorithm and its convergence analysis do not require any form of restricted isometry property on the model matrix $\mathbf{X}$.

Our proposed algorithm borrows ideas from projected gradient descent methods in first order convex optimization problems [45] and generalizes it to the discrete optimization Problem (20). We also derive new global convergence results for our proposed algorithms as presented in Theorem 3.1. Our proposal, with some novel modifications also applies to the non-smooth least absolute deviation loss with cardinality constraints as discussed in Section 3.3.

Consider the following optimization problem:

$$\min_{\boldsymbol{\beta}} \quad g(\boldsymbol{\beta}) \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq k, \tag{20}$$

where $g(\boldsymbol{\beta}) \geq 0$ is convex and has Lipschitz continuous gradient:

$$\|\nabla g(\boldsymbol{\beta}) - \nabla g(\widetilde{\boldsymbol{\beta}})\| \leq \ell \|\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}\|. \tag{21}$$

The first ingredient of our approach is the observation that when $g(\boldsymbol{\beta}) = \|\boldsymbol{\beta} - \mathbf{c}\|_2^2$ for a given $\mathbf{c}$, Problem (20) admits a closed form solution.

Proposition 3. *If $\hat{\boldsymbol{\beta}}$ is an optimal solution to the following problem:*

$$\hat{\boldsymbol{\beta}} \in \underset{\|\boldsymbol{\beta}\|_0 \leq k}{\arg\min} \quad \|\boldsymbol{\beta} - \mathbf{c}\|_2^2 \,, \tag{22}$$

then it can be computed as follows: $\hat{\boldsymbol{\beta}}$ retains the k largest (in absolute value) elements of $\mathbf{c} \in \mathbb{R}^p$ and sets the rest to zero, i.e., if $|c_{(1)}| \geq |c_{(2)}| \geq \ldots \geq |c_{(p)}|$, denote the ordered values of the absolute values of the vector $\mathbf{c}$, then:

$$\hat{\beta}_i = \begin{cases} c_i, & \text{if } i \in \{(1), \ldots, (k)\} \,, \\ 0, & \text{otherwise,} \end{cases} \tag{23}$$

where, $\hat{\beta}_i$ is the ith coordinate of $\hat{\boldsymbol{\beta}}$. We will denote the set of solutions to Problem (22) by the notation $\mathbf{H}_k(\mathbf{c})$.

Proof. We provide a proof of this in Section B.2, for the sake of completeness. □

Note that, we use the notation "argmin" (Problem (22) and in other places that follow) to denote the set of minimizers of the optimization Problem.

The operator (23) is also known as the hard-thresholding operator [20]—a notion that arises in the context of the following related optimization problem:

$$\hat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta}}{\arg\min} \quad \frac{1}{2}\|\boldsymbol{\beta} - \mathbf{c}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0, \tag{24}$$

where $\hat{\boldsymbol{\beta}}$ admits a simple closed form expression given by $\hat{\beta}_i = c_i$ if $|c_i| > \sqrt{\lambda}$ and $\hat{\beta}_i = 0$ otherwise, for $i = 1, \ldots, p$.

Remark 1. *There is an important difference between the minimizers of Problems (22) and (24). For Problem (24), the smallest (in absolute value) non-zero element in $\hat{\boldsymbol{\beta}}$ is greater than λ in absolute value. On the other hand, in Problem (22) there is no lower bound to the minimum (in absolute value) non-zero element of a minimizer. This needs to be taken care of while analyzing the convergence properties of Algorithm 1 (Section 3.2).*

Given a current solution $\boldsymbol{\beta}$, the second ingredient of our approach is to upper bound the function $g(\boldsymbol{\eta})$ around $g(\boldsymbol{\beta})$. To do so, we use ideas from projected gradient descent methods in first order convex optimization problems [45, 44].

Proposition 4. *([45, 44]) For a convex function $g(\boldsymbol{\beta})$ satisfying condition (21) and for any $L \geq \ell$ we have :*

$$g(\boldsymbol{\eta}) \leq Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) := g(\boldsymbol{\beta}) + \frac{L}{2}\|\boldsymbol{\eta} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta} - \boldsymbol{\beta} \rangle \tag{25}$$

for all $\boldsymbol{\beta}, \boldsymbol{\eta}$ with equality holding at $\boldsymbol{\beta} = \boldsymbol{\eta}$.

Applying Proposition 3 to the upper bound $Q_L(\boldsymbol{\eta}, \boldsymbol{\beta})$ in Proposition 4 we obtain

$$\begin{aligned}\underset{\|\boldsymbol{\eta}\|_0 \leq k}{\arg\min}\ Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) =& \underset{\|\boldsymbol{\eta}\|_0 \leq k}{\arg\min}\ \left(\frac{L}{2}\left\|\boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right)\right\|_2^2 - \frac{1}{2L}\|\nabla g(\boldsymbol{\beta})\|_2^2 + g(\boldsymbol{\beta})\right) \\ =& \underset{\|\boldsymbol{\eta}\|_0 \leq k}{\arg\min}\ \left\|\boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right)\right\|_2^2 \\ =& \mathbf{H}_k\left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right), \end{aligned} \tag{26}$$

where $\mathbf{H}_k(\cdot)$ is defined in (23). In light of (26) we are now ready to present Algorithm 1 to find a stationary point (see Definition 1) of Problem (20).

Algorithm 1

Input: $g(\boldsymbol{\beta})$, L, ϵ.
Output: A first order stationary solution $\boldsymbol{\beta}^*$.
Algorithm:

1. Initialize with $\boldsymbol{\beta}_1 \in \mathbb{R}^p$ such that $\|\boldsymbol{\beta}_1\|_0 \leq k$.
2. For $m \geq 1$, apply (26) with $\boldsymbol{\beta} = \boldsymbol{\beta}_m$ to obtain $\boldsymbol{\beta}_{m+1}$ as:

$$\boldsymbol{\beta}_{m+1} \in \mathbf{H}_k\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right) \tag{27}$$

3. Repeat Step 2, until $\|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2 \leq \epsilon$.
4. Let $\boldsymbol{\beta}_m := (\beta_{m1}, \ldots, \beta_{mp})$ denote the current estimate and let $I = \text{Supp}(\boldsymbol{\beta}_m) := \{i : \beta_{mi} \neq 0\}$. Solve the continuous optimization problem:

$$\min_{\boldsymbol{\beta}, \beta_i = 0,\ i \notin I} g(\boldsymbol{\beta}), \tag{28}$$

and let $\boldsymbol{\beta}^*$ be a minimizer.

The convergence properties of Algorithm 1 are presented in Section 3.2. We also present Algorithm 2, a variant of Algorithm 1 with better empirical performance. Algorithm 2 modifies Step 2 of Algorithm 1 by using a line search. It obtains $\boldsymbol{\eta}_m \in \mathbf{H}_k\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)$ and $\boldsymbol{\beta}_{m+1} = \lambda_m \boldsymbol{\eta}_m + (1-\lambda_m)\boldsymbol{\beta}_m$, where $\lambda_m \in \arg\min_\lambda g\left(\lambda \boldsymbol{\eta}_m + (1-\lambda)\boldsymbol{\beta}_m\right)$.

Note that the iterate $\boldsymbol{\beta}_m$ in Algorithm 2 need not be k-sparse (i.e., need not satisfy: $\|\boldsymbol{\beta}_m\|_0 \leq k$), however, $\boldsymbol{\eta}_m$ is k-sparse ($\|\boldsymbol{\eta}_m\|_0 \leq k$). Moreover, the sequence may not lead to a decreasing set of objective values, but it satisfies: $g(\boldsymbol{\beta}_{m+1}) \leq g(\boldsymbol{\eta}_m) \not\leq g(\boldsymbol{\beta}_m)$.

3.2 Convergence Analysis of Algorithm 1

In this section, we study convergence properties for Algorithm 1. Before we embark on the analysis, we need to define the notion of first order optimality for Problem (20).

Definition 1. *Given an $L \geq \ell$, the vector $\boldsymbol{\eta} \in \mathbb{R}^p$ is said to be a first order stationary point of Problem (20) if $\|\boldsymbol{\eta}\|_0 \leq k$ and it satisfies the following fixed point equation:*

$$\boldsymbol{\eta} \in \mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta}) \right). \tag{29}$$

Let us give some intuition associated with the above definition.

Consider $\boldsymbol{\eta}$ as in Definition 1. Since $\|\boldsymbol{\eta}\|_0 \leq k$, it follows that there is a set $I \subset \{1, \ldots, p\}$ such that $\eta_i = 0$ for all $i \in I$ and the size of I^c (complement of I) is k. Since $\boldsymbol{\eta} \in \mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta}) \right)$, it follows that for all $i \notin I$, we have: $\eta_i = \eta_i - \frac{1}{L} \nabla_i g(\boldsymbol{\eta})$, where, $\nabla_i g(\boldsymbol{\eta})$ is the ith coordinate of $\nabla g(\boldsymbol{\eta})$. It thus follows that: $\nabla_i g(\boldsymbol{\eta}) = 0$ for all $i \notin I$. Since $g(\boldsymbol{\eta})$ is convex in $\boldsymbol{\eta}$, this means that $\boldsymbol{\eta}$ solves the following convex optimization problem:

$$\min_{\boldsymbol{\eta}} \quad g(\boldsymbol{\eta}) \quad s.t. \quad \eta_i = 0, i \in I. \tag{30}$$

Note however, that the converse of the above statement is not true. That is, if $\tilde{I} \subset \{1, \ldots, p\}$ is an arbitrary subset with $|\tilde{I}^c| = k$ then a solution $\hat{\boldsymbol{\eta}}_{\tilde{I}}$ to the restricted convex problem (30) with $I = \tilde{I}$ need *not* correspond to a first order stationary point.

Note that any global minimizer to Problem (20) is also a first order stationary point, as defined above (see Proposition 7).

We present the following proposition (for its proof see Section B.6), which sheds light on a first order stationary point $\boldsymbol{\eta}$ for which $\|\boldsymbol{\eta}\|_0 < k$.

Proposition 5. *Suppose $\boldsymbol{\eta}$ satisfies the first order stationary condition (29) and $\|\boldsymbol{\eta}\|_0 < k$. Then $\boldsymbol{\eta} \in \arg\min_{\boldsymbol{\beta}} \quad g(\boldsymbol{\beta})$.*

We next define the notion of an ϵ-approximate first order stationary point of Problem (20):

Definition 2. *Given an $\epsilon > 0$ and $L \geq \ell$ we say that $\boldsymbol{\eta}$ satisfies an ϵ-approximate first order optimality condition of Problem (20) if $\|\boldsymbol{\eta}\|_0 \leq k$ and for some $\hat{\boldsymbol{\eta}} \in \mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta}) \right)$, we have $\|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}\|_2 \leq \epsilon$.*

Before we dive into the convergence properties of Algorithm 1, we need to introduce some notation. Let $\boldsymbol{\beta}_m = (\beta_{m1}, \ldots, \beta_{mp})$ and $\mathbf{1}_m = (e_1, \ldots, e_p)$ with $e_j = 1$, if $\beta_{mj} \neq 0$, and $e_j = 0$, if $\beta_{mj} = 0$, $j = 1, \ldots, p$, i.e., $\mathbf{1}_m$ represents the sparsity pattern of the support of $\boldsymbol{\beta}_m$.

Suppose, we order the coordinates of $\boldsymbol{\beta}_m$ by their absolute values: $|\beta_{(1),m}| \geq |\beta_{(2),m}| \geq \ldots \geq |\beta_{(p),m}|$. Note that by definition (27), $\beta_{(i),m} = 0$ for all $i > k$ and $m \geq 2$. We denote $\alpha_{k,m} = |\beta_{(k),m}|$ to be the kth largest (in absolute value) entry in $\boldsymbol{\beta}_m$ for all $m \geq 2$. Clearly if $\alpha_{k,m} > 0$ then $\|\boldsymbol{\beta}_m\|_0 = k$ and if $\alpha_{k,m} = 0$ then $\|\boldsymbol{\beta}_m\|_0 < k$. Let $\overline{\alpha}_k := \limsup_{m\to\infty} \alpha_{k,m}$ and $\underline{\alpha}_k := \liminf_{m\to\infty} \alpha_{k,m}$.

Proposition 6. *Consider $g(\boldsymbol{\beta})$ and ℓ as defined in (20) and (21). Let $\boldsymbol{\beta}_m, m \geq 1$ be the sequence generated by Algorithm 1. Then we have :*

(a) *For any $L \geq \ell$, the sequence $g(\boldsymbol{\beta}_m)$ satisfies*

$$g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1}) \geq \frac{L-\ell}{2} \left\|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\right\|_2^2, \tag{31}$$

is decreasing and converges.

(b) *If $L > \ell$, then $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to \mathbf{0}$ as $m \to \infty$.*

(c) *If $L > \ell$ and $\underline{\alpha}_k > 0$ then the sequence $\mathbf{1}_m$ converges after finitely many iterations, i.e., there exists an iteration index M^* such that $\mathbf{1}_m = \mathbf{1}_{m+1}$ for all $m \geq M^*$. Furthermore, the sequence $\boldsymbol{\beta}_m$ is bounded and converges to a first order stationary point.*

(d) *If $L > \ell$ and $\underline{\alpha}_k = 0$ then $\liminf_{m\to\infty} \|\nabla g(\boldsymbol{\beta}_m)\|_\infty = 0$.*

(e) *Let $L > \ell$, $\overline{\alpha}_k = 0$ and suppose that the sequence $\boldsymbol{\beta}_m$ has a limit point. Then $g(\boldsymbol{\beta}_m) \to \min_{\boldsymbol{\beta}} g(\boldsymbol{\beta})$.*

Proof. See Section B.1. □

Remark 2. *Note that the existence of a limit point in Proposition 6, Part (e) is guaranteed under fairly weak conditions. One such condition is that $\sup(\{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \leq k, f(\boldsymbol{\beta}) \leq f_0\}) < \infty$, for any finite value f_0. In words this means that the k-sparse level sets of the function $g(\boldsymbol{\beta})$ is bounded.*

In the special case where $g(\boldsymbol{\beta})$ is the least squares loss function, the above condition is equivalent to every k-submatrix $(\mathbf{X}_J)$ of $\mathbf{X}$ comprising of k columns being full rank. In particular, this holds with probability one when the entries of $\mathbf{X}$ are drawn from a continuous distribution and $k < n$.

Remark 3. *Parts (d) and (e) of Proposition 6 are probably not statistically interesting cases, since they correspond to un-regularized solutions of the problem $\min g(\boldsymbol{\beta})$. However, we include them since they shed light on the properties of Algorithm 1.*

The conditions assumed in Part (c) imply that the support of $\boldsymbol{\beta}_m$ stabilizes and Algorithm 1 behaves like vanilla gradient descent thereafter. The support of $\boldsymbol{\beta}_m$ need not

stabilize for Parts (d), (e) and thus Algorithm 1 may not behave like vanilla gradient descent after finitely many iterations. However, the objective values (under minor regularity assumptions) converge to $\min\ g(\boldsymbol{\beta})$.

We present the following Proposition (for proof see Section B.3) about a uniqueness property of the fixed point equation (1).

Proposition 7. *Suppose* $L > \ell$ *and let* $\boldsymbol{\eta}$ *satisfy a first order stationary point as in Definition 1. Then the set* $\mathbf{H}_k\left(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta})\right)$ *has exactly one element:* $\boldsymbol{\eta}$.

The following proposition (for a proof see Section B.4) shows that a global minimizer of the Problem (20) is also a first order stationary point.

Proposition 8. *Suppose* $L > \ell$ *and let* $\widehat{\boldsymbol{\beta}}$ *be a global minimizer of Problem (20). Then* $\widehat{\boldsymbol{\beta}}$ *is a first order stationary point.*

Proposition 6 establishes that Algorithm 1 either converges to a first order stationarity point (part (c)) or it converges[1] to a global optimal solution (Parts (d), (e)), but does not quantify the rate of convergence. We next characterize the rate of convergence of the algorithm to an ϵ-approximate first order stationary point.

Theorem 3.1. *Let* $L > \ell$ *and* $\boldsymbol{\beta}^*$ *denote a first order stationary point of Algorithm 1. After* M *iterations Algorithm 1 satisfies*

$$\min_{m=1,\ldots,M} \|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2 \leq \frac{2(g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}^*))}{M(L-\ell)}, \tag{32}$$

where $g(\boldsymbol{\beta}_m) \downarrow g(\boldsymbol{\beta}^*)$ *as* $m \to \infty$.

Proof. See Section B.5. □

Theorem 3.1 implies that for any $\epsilon > 0$ there exists $M = O(\frac{1}{\epsilon})$ such that for some $1 \leq m^* \leq M$, we have: $\|\boldsymbol{\beta}_{m^*+1} - \boldsymbol{\beta}_{m^*}\|_2^2 \leq \epsilon$. Note that the convergence rates derived above apply for a large class of problems (20), where, the function $g(\boldsymbol{\beta}) \geq 0$ is convex with Lipschitz continuous gradient (21). Tighter rates may be obtained under additional structural assumptions on $g(\cdot)$. For example, the adaptation of Algorithm 1 for Problem (4) was analyzed in [8, 9] with $\mathbf{X}$ satisfying coherence [8] or Restricted Isometry Property (RIP) [9]. In these cases, the algorithm can be shown to have a linear convergence rate [8, 9], where the rate depends upon the RIP constants.

Note that by Proposition 6 the support of $\boldsymbol{\beta}_m$ stabilizes after finitely many iterations, after which Algorithm 1 behaves like gradient descent on the stabilized support. If $g(\boldsymbol{\beta})$ restricted to this support is strongly convex, then Algorithm 1 will enjoy a linear rate of convergence [45], as soon as the support stabilizes. This behavior is adaptive, i.e., Algorithm 1 does not need to be modified after the support stabilizes.

[1] under minor technical assumptions

The next section describes practical post-processing schemes via which first order stationary points of Algorithm 1 can be obtained by solving a low dimensional convex optimization problem, as soon as the support is found to stabilize, numerically. In our numerical experiments, we this version of Algorithm 1 (with multiple starting points) took at most a few minutes for $p = 2000$ and a few seconds for smaller values of p.

Polishing coefficients on the active set

Algorithm 1 *detects* the active set after a few iterations. Once the active set stabilizes, the algorithm may take a number of iterations to estimate the values of the regression coefficients on the active set to a high accuracy level.

In this context, we found the following simple polishing of coefficients to be useful. When the algorithm has converged to a tolerance of ϵ ($\approx 10^{-4}$), we fix the current active set, $\mathcal{I}$, and solve the following lower-dimensional convex optimization problem:

$$\min_{\boldsymbol{\beta}, \beta_i = 0, i \notin \mathcal{I}} \quad g(\boldsymbol{\beta}). \tag{33}$$

In the context of the least squares and the least absolute deviation problems, Problem (33) reduces to to a smaller dimensional least squares and a linear optimization problem respectively, which can be solved very efficiently up to a very high level of accuracy.

3.3 Application to Least Squares

For the support constrained problem with squared error loss, we have $g(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ and $\nabla g(\boldsymbol{\beta}) = -\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$. The general algorithmic framework developed above applies in a straightforward fashion for this special case. Note that for this case $\ell = \lambda_{\max}(\mathbf{X}'\mathbf{X})$.

The polishing of the regression coefficients in the active set can be performed via a least squares problem on $\mathbf{y}, \mathbf{X}_I$, where I denotes the support of the regression coefficients.

3.4 Application to Least Absolute Deviation

We will now show how the method proposed in the previous section applies to the least absolute deviation problem with support constraints in $\boldsymbol{\beta}$:

$$\min_{\boldsymbol{\beta}} \quad g_1(\boldsymbol{\beta}) := \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_1 \;\; s.t. \;\; \|\boldsymbol{\beta}\|_0 \leq k. \tag{34}$$

Since $g_1(\boldsymbol{\beta})$ is non-smooth, our framework does not apply directly. We smooth the non-differentiable $g_1(\boldsymbol{\beta})$ so that we can apply Algorithms 1 and 2. Observing that $g_1(\boldsymbol{\beta}) = \sup_{\|\mathbf{w}\|_\infty \leq 1} \langle \mathbf{Y} - X\boldsymbol{\beta}, \mathbf{w} \rangle$ we make use of the smoothing technique of [43] to obtain $g_1(\boldsymbol{\beta}; \tau) = \sup_{\|\mathbf{w}\|_\infty \leq 1} (\langle \mathbf{Y} - X\boldsymbol{\beta}, \mathbf{w} \rangle - \frac{\tau}{2}\|\mathbf{w}\|_2^2)$; which is a smooth approximation of $g_1(\beta)$, with $\ell = \frac{\lambda_{\max}(\mathbf{X}'\mathbf{X})}{\tau}$ for which Algorithms 1 and 2 apply.

In order to obtain a good approximation to Problem (34), we found the following strategy to be useful in practice:

1. Fix $\tau > 0$, initialize with $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ and repeat the following steps [2]—[3] till convergence:
2. Apply Algorithm 1 (or Algorithm 2) to the smooth function $g_1(\boldsymbol{\beta}; \tau)$. Let $\boldsymbol{\beta}^*_\tau$ be the limiting solution.
3. Decrease $\tau \leftarrow \tau\gamma$ for some pre-defined constant $\gamma = 0.8$ (say), and go back to step [1] with $\boldsymbol{\beta}_0 = \boldsymbol{\beta}^*_\tau$. Exit if $\tau <$ TOL, for some pre-defined tolerance.

4 A Brief Tour of the Statistical Properties of Problem (1)

As already alluded to in the introduction, there is a substantial body of impressive work characterizing the theoretical properties of best subset solutions in terms of various metrics: predictive performance, estimation of regression coefficients, and variable selection properties. For the sake of completeness, we present a brief review of some of the properties of solutions to Problem (1) in Section C.

5 Computational Experiments for Subset Selection with Least Squares Loss

In this section, we present a variety of computational experiments to assess the algorithmic and statistical performances of our approach. We consider both the classical overdetermined case with $n > p$ (Section 5.2) and the high dimensional $p \gg n$ case (Section 5.3) for the least squares loss function with support constraints.

5.1 Description of Experimental Data

We demonstrate the performance of our proposal via a series of experiments on both synthetic and real data.

Synthetic Datasets. We consider a collection of problems where $\mathbf{x}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma}), i = 1, \ldots, n$ are independent realizations from a p-dimensional multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Sigma} := (\sigma_{ij})$. The columns of the $\mathbf{X}$ matrix were subsequently standardized to have unit ℓ_2 norm. For a fixed $\mathbf{X}_{n\times p}$, we generated the response $\mathbf{y}$ as follows: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \boldsymbol{\epsilon}$, where $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$. We denote the number of nonzeros in $\boldsymbol{\beta}^0$ by k_0. The choice of $\mathbf{X}, \boldsymbol{\beta}^0, \sigma$ determines the Signal-to-Noise Ratio (SNR) of the problem, which is defined as: $\text{SNR} = \frac{\text{var}(\mathbf{x}'\boldsymbol{\beta}^0)}{\sigma^2}$.

We considered the following four different examples:

Example 1: We took $\sigma_{ij} = \rho^{|i-j|}$ for $i, j \in \{1, \ldots, p\} \times \{1, \ldots, p\}$. We consider different values of $k_0 \in \{5, 10\}$ and $\beta_i^0 = 1$ for k_0 equi-spaced values. In the case where exactly equi-spaced values are not possible we rounded the indices to the nearest large integer value. of i in the range $\{1, 2, \ldots, p\}$.

Example 2: We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 5$ and $\boldsymbol{\beta}^0 = (\mathbf{1}'_{5\times 1}, \mathbf{0}'_{p-5\times 1})' \in \mathbb{R}^p$.

Example 3: We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 10$ and $\beta_i^0 = \frac{1}{2} + (10 - \frac{1}{2})\frac{(i-1)}{k_0}, i = 1, \ldots, 10$ and $\beta_i^0 = 0, \forall i > 10$ — i.e., a vector with ten nonzero entries, with the nonzero values being equally spaced in the interval $[\frac{1}{2}, 10]$.

Example 4: We took $\mathbf{\Sigma} = \mathbf{I}_{p\times p}$, $k_0 = 6$ and $\boldsymbol{\beta}^0 = (-10, -6, -2, 2, 6, 10, \mathbf{0}_{p-6})$, i.e., a vector with six nonzero entries, equally spaced in the interval $[-10, 10]$.

Real Datasets We considered the Diabetes dataset analyzed in [23]. We used the dataset with all the second order interactions included in the model, which resulted in 64 predictors. We reduced the sample size to $n = 350$ by taking a random sample and standardized the response and the columns of the model matrix to have zero means and unit ℓ_2-norm.

In addition to the above, we also considered a real microarray dataset: the Leukemia data [18]. We downloaded the processed dataset from http://stat.ethz.ch/~dettling/bagboost.html, which had $n = 72$ binary responses and more than 3000 predictors. We standardized the response and columns of features to have zero means and unit ℓ_2-norm. We reduced the set of features to 1000 by retaining the features maximally correlated (in absolute value) to the response. We call the resulting feature matrix $\mathbf{X}_{n\times p}$ with $n = 72, p = 1000$. We then generated a semi-synthetic dataset with continuous

response as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \epsilon$, where the first five coefficients of $\boldsymbol{\beta}^0$ were taken as one and the rest as zero. The noise was distributed as $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$, with σ^2 chosen to get a SNR=7.

Computer Specifications and Software Computations were carried out in a linux 64 bit server—Intel(R) Xeon(R) eight-core processor @ 1.80GHz, 16 GB of RAM for the overdetermined $n > p$ case and in a Dell Precision T7600 computer with an Intel Xeon E52687 sixteen-core processor @ 3.1GHz, 128GB of Ram for the high-dimensional $p \gg n$ case. The discrete first order methods were implemented in MATLAB 2012b. We used GUROBI [33] version 5.5 and the MATLAB interface to GUROBI for all of our experiments, apart from the computations for synthetic data for $n > p$, which were done in GUROBI via its Python 2.7 interface.

5.2 The Overdetermined Regime: $n > p$

Using the Diabetes dataset and synthetic datasets, we demonstrate the combined effect of using the discrete first order methods with the MIO approach. Together, these methods show improvements in obtaining good upper bounds and in closing the MIO gap to certify global optimality. Using synthetic datasets where we know the true linear regression model, we perform side-by-side comparisons of this method with several other state-of-the-art algorithms designed to estimate sparse linear models.

5.2.1 Obtaining Good Upper Bounds

We conducted experiments to evaluate the performance of our methods in terms of obtaining high quality solutions for Problem (1).

We considered the following three algorithms:

(a) Algorithm 2 with fifty random initializations[2]. We took the solution corresponding to the best objective value.

(b) MIO with cold start, i.e., formulation (9) with a time limit of 500 seconds.

(c) MIO with warm start. This was the MIO formulation initialized with the discrete first order optimization solution obtained from **(a)**. This was run for a total of 500 seconds.

[2]we took fifty random starting values around $\mathbf{0}$ of the form $\min(i-1, 1)\epsilon, i = 1, \ldots, 50$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}_{p\times 1}, 4\mathbf{I})$. We found empirically that Algorithm 2 provided better upper bounds than Algorithm 1.

To compare the different algorithms in terms of the quality of upper bounds, we run for every instance all the algorithms and obtain the best solution among them, say, f_*. If f_{alg} denotes the value of the best subset objective function for method "alg", then we define the relative accuracy of the solution obtained by "alg" as:

$$\text{Relative Accuracy} = (f_{\text{alg}} - f_*)/f_*, \tag{35}$$

where alg $\in \{(\text{a}), (\text{b}), (\text{c})\}$ as described above.

We did experiments for the Diabetes dataset for different values of k (see Table 1). For each of the algorithms we report the amount of time taken by the algorithm to reach the best objective value during the time of 500 seconds.

k	Discrete First Order		MIO Cold Start		MIO Warm Start	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
9	0.1306	1	0.0036	500	0	346
20	0.1541	1	0.0042	500	0	77
49	0.1915	1	0.0015	500	0	87
57	0.1933	1	0	500	0	2

Table 1: Quality of upper bounds for Problem (1) for the Diabetes dataset, for different values of k. We see that the MIO equipped with warm starts deliver the best upper bounds in the shortest overall times. The run time for the MIO with warm start includes the time taken by the discrete first order method (which were all less than a second).

Using the discrete first order methods in combination with the MIO algorithm resulted in finding the best possible relative accuracy in a matter of a few minutes.

5.2.2 Improving MIO Performance via Warm Starts

We performed a series of experiments on the Diabetes dataset to obtain a globally optimal solution to Problem (1) via our approach and to understand the implications of using advanced warm starts to the MIO formulation in terms of certifying optimality. For each choice of k we ran Algorithm 2 with fifty random initializations. They took less than a few seconds to run. We used the best solution as an advanced warm start to the MIO formulation (9). For each of these examples, we also ran the MIO formulation without any warm start information and also without the parameter specifications in Section 2.3 (we refer to this as "cold start"). Figure 3 summarizes the results. The figure shows that in the presence of warm starts and problem specific side information, the MIO closes the optimality gap significantly faster.

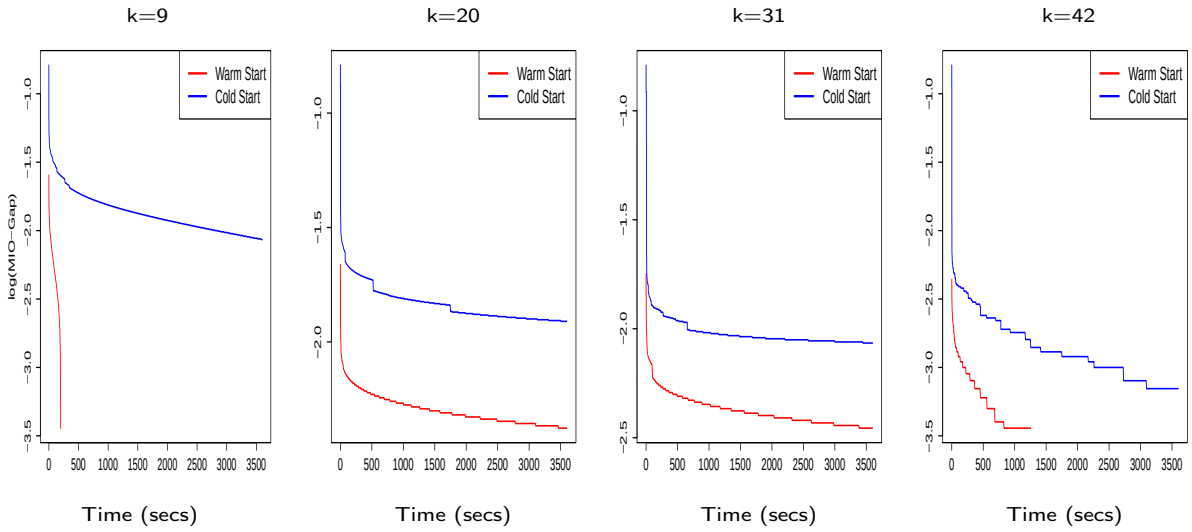

Figure 3: The evolution of the MIO optimality gap (in $\log_{10}(\cdot)$ scale) for Problem (1), for the Diabetes dataset with $n = 350, p = 64$ with and without warm starts (and parameter specifications as in Section 2.3) for different values of k. The MIO significantly benefits by advanced warm starts delivered by Algorithm 2. In all of these examples, the global optimum was found within a very small fraction of the total time, but the proof of global optimality came later.

5.2.3 Statistical Performance

We considered datasets as described in Example 1, Section 5.1—we took different values of n, p with $n > p$, ρ with $k_0 = 10$.

Competing Methods and Performance Measures For every example, we considered the following learning procedures for comparison purposes: (a) the MIO approach equipped warm starts from Algorithm 2 (annotated as "MIO" in the figure), (b) the `Lasso`, (c) `Sparsenet` and (d) stepwise regression (annotated as "Step" in the figure).

We used `R` to compute `Lasso`, `Sparsenet` and stepwise regression using the glmnet 1.7.3, `Sparsenet` and Stats 3.0.2 packages respectively, which were all downloaded from `CRAN` at `http://cran.us.r-project.org/`.

In addition to the above, we have also performed comparisons with a *debiased* version of the `Lasso`: i.e., performing unrestricted least squares on the `Lasso` support to mitigate the bias imparted by `Lasso` shrinkage.

We note that `Sparsenet` [38] considers a penalized likelihood formulation of the form (3),

where the penalty is given by the generalized MCP penalty family (indexed by λ, γ) for a family of values of $\gamma \geq 1$ and $\lambda \geq 0$. The family of penalties used by `Sparsenet` is thus given by: $p(t;\gamma;\lambda) = \lambda(|t| - \frac{t^2}{2\lambda\gamma})\mathbf{I}(|t| < \lambda\gamma) + \frac{\lambda^2\gamma}{2}\mathbf{I}(|t| \geq \lambda\gamma)$ for γ, λ described as above. As $\gamma = \infty$ with λ fixed, we get the penalty $p(t;\gamma;\lambda) = \lambda|t|$. The family above includes as a special case ($\gamma = 1$), the hard thresholding penalty, a penalty recommended in the paper [60] for its useful statistical properties.

For each procedure, we obtained the "optimal" tuning parameter by selecting the model that achieved the best predictive performance on a held out validation set. Once the model $\widehat{\boldsymbol{\beta}}$ was selected, we obtained the prediction error as:

$$\text{Prediction Error} = \|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^0\|_2^2 / \|\mathbf{X}\boldsymbol{\beta}^0\|_2^2. \tag{36}$$

We report "prediction error" and number of non-zeros in the optimal model in our results. The results were averaged over ten random instances, for different realizations of $\mathbf{X}, \epsilon$. For every run: the training and validation data had a fixed $\mathbf{X}$ but random noise ϵ.

Figure 4 presents results for data generated as per Example 1 with $n = 500$ and $p = 100$. We see that the MIO procedure performs very well across all the examples. Among the methods, MIO performs the best, followed by `Sparsenet`, `Lasso` with Step(wise) exhibiting the worst performance. In terms of prediction error, the MIO performs the best, only to be marginally outperformed by `Sparsenet` in a few instances. This further illustrates the importance of using non-convex methods in sparse learning. Note that the MIO approach, unlike `Sparsenet` certifies global optimality in terms of solving Problem 1. However, based on the plots in the upper panel, `Sparsenet` selects a few redundant variables unlike MIO. `Lasso` delivers quite dense models and pays the price in predictive performance too, by selecting wrong variables. As the value of SNR increases, the predictive power of the methods improve, as expected. The differences in predictive errors between the methods diminish with increasing SNR values. With increasing values of ρ (from left panel to right panel in the figure), the number of non-zeros selected by the `Lasso` in the optimal model increases.

We also performed experiments with the debiased version of the `Lasso`. The unrestricted least squares solution on the optimal model selected by `Lasso` (as shown in Figure 4) had worse predictive performance than the `Lasso`, with the same sparsity pattern. This is probably due to overfitting since the model selected by the `Lasso` is quite dense compared to n, p. We also tried some variants of debiased `Lasso` which led to models with better performances than the `Lasso` but the results were inferior compared to MIO — we provide a detailed description in Section D.2.

We also performed experiments with $n = 1000, p = 50$ for data generated as per Example 1. We solved the problems to provable optimality and found that the MIO

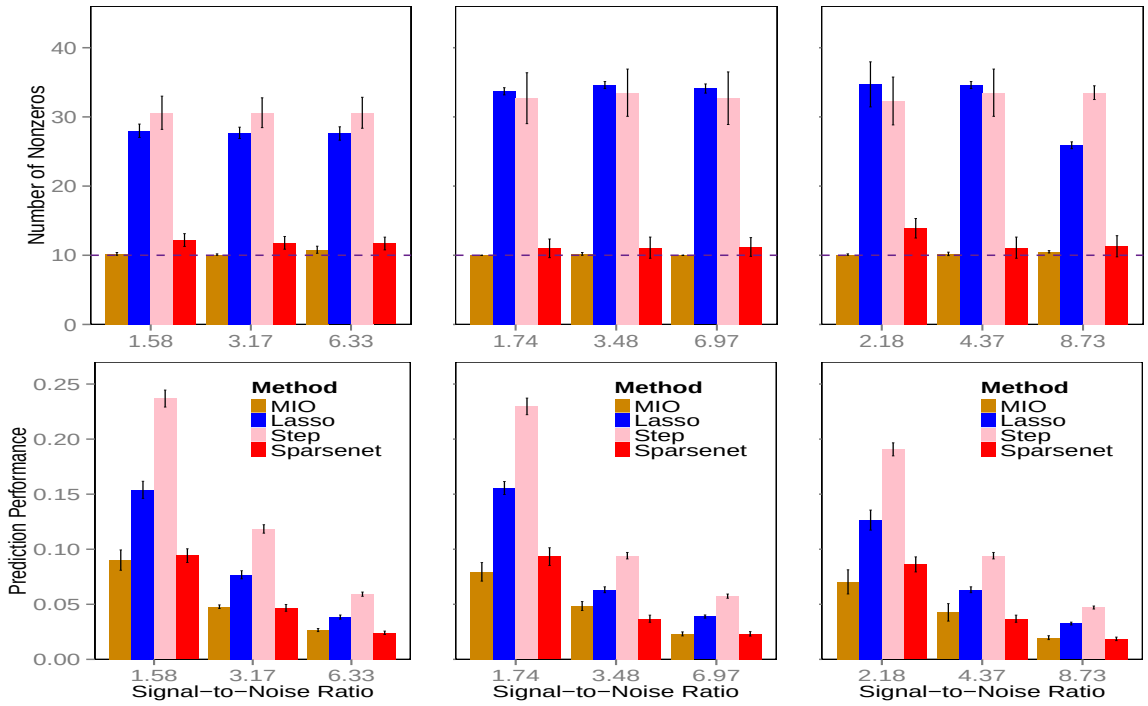

Figure 4: Figure showing the sparsity (upper panel) and predictive performances (bottom panel) for different subset selection procedures for the least squares loss. Here, we consider data generated as per Example 1, with $n = 500, p = 100$, $k_0 = 10$, for three different SNR values with [Left Panel] $\rho = 0.5$, [Middle Panel] $\rho = 0.8$, and [Right Panel] $\rho = 0.9$. The dashed line in the top panel represents the true number of nonzero values. For each of the procedures, the optimal model was selected as the one which produced the best prediction accuracy on a separate validation set, as described in Section 5.2.3.

performed very well when compared to other competing methods. We do not report the experiments for brevity.

5.2.4 MIO model training

We trained a sequence of best subset models (indexed by k) by applying the MIO approach with warm starts. Instead of running the MIO solvers from scratch for different values of k, we used *callbacks*, a feature of integer optimization solvers. Callbacks allow the user to solve an initial model, and then add additional constraints to the model one at a time. These "cuts" reduce the size of the feasible region without having to rebuild the entire optimization model. Thus, in our case, we can save time by building the initial optimization model for $k = p$. Once the solution for $k = p$ is obtained, a cut can be added to the model: $\sum_{i=1}^{p} z_i \leq k$ for $k = p - 1$ and the model can be re-solved

from this point. We apply this procedure until we arrive at a model with $k = 1$.

For each value of k tested, the MIO best subset algorithm was set to stop the first time either an optimality gap of 1% was reached or a time limit of 15 minutes was reached. Additionally, we only tested values of k from 5 through 25, and used Algorithm 2 to warm start the MIO algorithm. We observed that it was possible to obtain speedups of a factor of 2-4 by carefully tuning the optimization solver for a particular problem, but chose to maintain generality by solving with default parameters. Thus, we do not report times with the intention of accurately benchmarking the best possible time but rather to show that it is computationally tractable to solve problems to optimality using modern MIO solvers.

5.3 The High-Dimensional Regime: $p \gg n$

In this section, we investigate **(a)** the evolution of upper bounds in the high-dimensional regime, **(b)** the effect of a bounding box formulation on the speed of closing the optimality gap and **(c)** the statistical performance of the MIO approach in comparison to other state-of-the art methods.

5.3.1 Obtaining Good Upper Bounds

We performed tests similar to those in Section 5.2.1 for the $p \gg n$ regime. We tested a synthetic dataset corresponding to Example 2 with $n = 30, p = 2000$ for varying SNR values (see Table 2) over a time of 500s. As before, using the discrete first order methods in combination with the MIO algorithm resulted in finding the best possible upper bounds in the shortest possible times.

We also did experiments on the Leukemia dataset. In Figure 5 we demonstrate the evolution of the objective value of the best subset problem for different values of k. For each value of k, we warm-started the MIO with the solution obtained by Algorithm 2 and allowed the MIO solver to run for 4000 seconds. The best objective value obtained at the end of 4000 seconds is denoted by f_*. We plot the Relative Accuracy, i.e., $(f_t - f_*)/f_*$, where f_t is the objective value obtained after t seconds. The figure shows that the solution obtained by Algorithm 2 is improved by the MIO on various instances and the time taken to improve the upper bounds depends upon k. In general, for smaller values of k the upper bounds obtained by the MIO algorithm stabilize earlier, i.e., the MIO finds improved solutions faster than larger values of k.

	k	Discrete First Order		MIO Cold Start		MIO Warm Start	
		Accuracy	Time	Accuracy	Time	Accuracy	Time
SNR = 3	5	0.1647	37.2	1.0510	500	0	72.2
	6	0.6152	41.1	0.2769	500	0	77.1
	7	0.7843	40.7	0.8715	500	0	160.7
	8	0.5515	38.8	2.1797	500	0	295.8
	9	0.7131	45.0	0.4204	500	0	96.0
SNR = 7	5	0.5072	45.6	0.7737	500	0	65.6
	6	1.3221	40.3	0.5121	500	0	82.3
	7	0.9745	40.9	0.7578	500	0	210.9
	8	0.8293	40.5	1.8972	500	0	262.5
	9	1.1879	44.2	0.4515	500	0	254.2

Table 2: The quality of upper bounds for Problem (1) obtained by Algorithm 2, MIO with cold start and MIO warm-started with Algorithm 2. We consider the synthetic dataset of Example 2 with $n = 30, p = 2000$ and different values of SNR. The MIO method, when warm-started with the first order solution performs the best in terms of getting a good upper bound in the shortest time. The metric "Accuracy" is defined in (35). The first order methods are fast but need not lead to highest quality solutions on their own. MIO improves the quality of upper bounds delivered by the first order methods and their combined effect leads to the best performance.

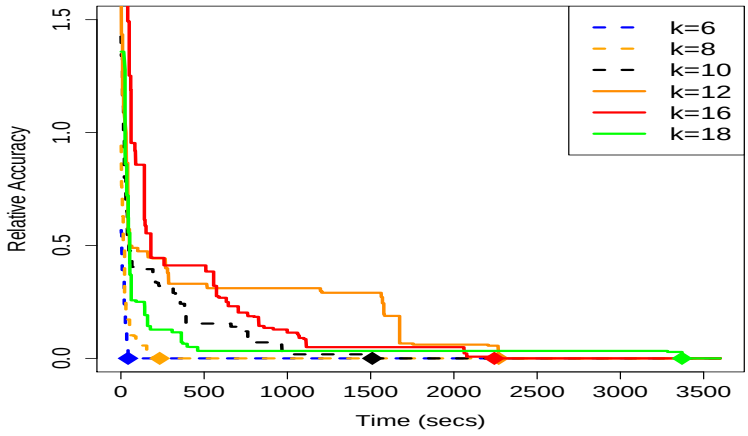

Figure 5: Behavior of MIO aided with warm start in obtaining good upper bounds over time for the Leukemia dataset ($n = 72, p = 1000$). The vertical axis shows relative accuracy, i.e., $(f_t - f_*)/f_*$, where f_t is the objective value obtained after t seconds and f_* denotes the best objective value obtained by the method after 4000 seconds. The colored diamonds correspond to the locations where the MIO (with warm start) attains the best solution. The figure shows that MIO improves the solution obtained by the first order method in all the instances. The time at which the best possible upper bound is obtained depends upon the choice of k. Typically larger k values make the problem harder—hence the best solutions are obtained after a longer wait.

5.3.2 Bounding Box Formulation

With the aid of advanced warm starts as provided by Algorithm 2, the MIO obtains a very high quality solution very quickly—in most of the examples the solution thus obtained turns out to be the global minimum. However, in the typical "high-dimensional" regime, with $p \gg n$, we observe that the certificate of global optimality comes later as the lower bounds of the problem "evolve" slowly. This is observed even in the presence of warm starts and using the implied bounds as developed in Section 2.2 and is aggravated for the cold-started MIO formulation (10).

To address this, we consider the MIO formulation (37) obtained by adding bounding boxes around a local solution. These restrictions *guide* the MIO in restricting its *search* space and enable the MIO to certify global optimality inside that bounding box. We consider the following additional bounding box constraints to the MIO formulation (10):

$$\left\{\boldsymbol{\beta} : \|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}_0\|_1 \leq \mathcal{L}^{\zeta}_{\ell,\text{loc}}\right\} \cap \left\{\boldsymbol{\beta} : \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \leq \mathcal{L}^{\beta}_{\ell,\text{loc}}\right\},$$

where, $\boldsymbol{\beta}_0$ is a candidate sparse solution. The radii of the two ℓ_1-balls above, namely, $\mathcal{L}^{\zeta}_{\ell,\text{loc}}$ and $\mathcal{L}^{\beta}_{\ell,\text{loc}}$ are user-defined parameters and control the size of the feasible set.

Using the notation $\boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}$ we have the following MIO formulation (equipped with the additional bounding boxes):

$$\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{z},\boldsymbol{\zeta}} \quad & \tfrac{1}{2}\,\boldsymbol{\zeta}^T\boldsymbol{\zeta} - \langle \mathbf{X}'\mathbf{y}, \boldsymbol{\beta}\rangle + \tfrac{1}{2}\,\|\mathbf{y}\|_2^2 \\
s.t. \quad & \boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta} \\
& (\beta_i, 1 - z_i) : \text{SOS type-1}, \;\; i = 1, \ldots, p \\
& z_i \in \{0, 1\}, i = 1, \ldots, p \\
& \sum_{i=1}^{p} z_i \leq k \\
& -\mathcal{M}_U \leq \beta_i \leq \mathcal{M}_U, i = 1, \ldots, p \\
& \|\boldsymbol{\beta}\|_1 \leq \mathcal{M}_\ell \\
& -\mathcal{M}^{\zeta}_U \leq \zeta_i \leq \mathcal{M}^{\zeta}_U, i = 1, \ldots, n \\
& \|\boldsymbol{\zeta}\|_1 \leq \mathcal{M}^{\zeta}_\ell \\
& \|\boldsymbol{\zeta} - \boldsymbol{\zeta}_0\|_1 \leq \mathcal{L}^{\zeta}_{\ell,\text{loc}} \\
& \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \leq \mathcal{L}^{\beta}_{\ell,\text{loc}}.
\end{aligned} \tag{37}$$

For large values of $\mathcal{L}^{\zeta}_{\ell,\text{loc}}$ (respectively, $\mathcal{L}^{\beta}_{\ell,\text{loc}}$) the constraints on $\mathbf{X}\boldsymbol{\beta}$ (respectively, $\boldsymbol{\beta}$) become ineffective and one gets back formulation (10). To see the impact of these additional cutting planes in the MIO formulation, we consider a few examples as illustrated in Figures 6,7,12.

Interpretation of the bounding boxes A local bounding box in the variable $\boldsymbol{\zeta} = \mathbf{X}\boldsymbol{\beta}$ directs the MIO solver to seek for candidate solutions that deliver models with predictive accuracy "similar" (controlled by the radius of the ball) to a reference predictive model, given by $\boldsymbol{\zeta}_0$. In our experiments, we typically chose $\boldsymbol{\zeta}_0$ as the solution delivered by running MIO (warm-started with a first order solution) for a few hundred to a few thousand seconds. More generally, $\boldsymbol{\zeta}_0$ may be selected by any other sparse learning method. In our experiments, we found that the run-time behavior of the MIO depends upon how correlated the columns of $\mathbf{X}$ are — more correlation leading to longer run-times.

Similarly, a bounding box around $\boldsymbol{\beta}$ directs the MIO to look for solutions in the neighborhood of a reference point $\boldsymbol{\beta}_0$. In our experiments, we chose the reference $\boldsymbol{\beta}_0$ as the solution obtained by MIO (warm-started with a first order solution) and allowing it to run for a few hundred to a few thousand seconds. We observed that the MIO solver in presence of bounding boxes in the $\boldsymbol{\beta}$-space certified optimality and in the process finding better solutions; much faster than the $\boldsymbol{\zeta}$-bounding box method.

Note that the $\boldsymbol{\beta}$-bounding box constraint leads to $O(p)$ and the $\boldsymbol{\zeta}$-box leads to $O(n)$ constraints. Thus, when $p \gg n$ the additional $\boldsymbol{\zeta}$ constraints add a fewer number of extra variables when compared to the $\boldsymbol{\beta}$ constraints.

Experiments In the first set of experiments, we consider the Leukemia dataset with $n = 72, p = 1000$. We took two different values of $k \in \{5, 10\}$ and for each case we ran Algorithm 2 with several random restarts. The best solution thus obtained was used to warm start the MIO formulation (10), which we ran for an additional 3600 seconds. The solution thus obtained is denoted by $\boldsymbol{\beta}_0$. We then consider formulation (37) with $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$ and different values of $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \text{Frac}$ (as annotated in Figure 6) — the results are displayed in Figure 6.

We consider another set of experiments to demonstrate the performance of the MIO in certifying global optimality for different synthetic datasets with varying n, p, k as well as with different structures on the bounding box. In the first case, we generated data as per Example 1 with $\rho = 0.9$, $k_0 = 5$. We consider the case with $\boldsymbol{\zeta}_0 = \mathbf{X}\boldsymbol{\beta}_0$, $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \infty$ and $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = 0.5\|\mathbf{X}\boldsymbol{\beta}_0\|_1$, where $\boldsymbol{\beta}_0$ is a k-sparse solution obtained from the MIO formulation (10) run with a time limit of 1000 seconds, after being warm-started with Algorithm 2. The results are displayed in Figure 7[Left Panel]. In the second case (with data same as before) we obtained $\boldsymbol{\beta}_0$ in the same fashion as described before—we took a bounding box around $\boldsymbol{\beta}_0$, and left the box constraint around $\mathbf{X}\boldsymbol{\beta}_0$ inactive, i.e., we set $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$ and $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \|\boldsymbol{\beta}_0\|_1/k$. We performed two sets of experiments, where the data were generated based on different SNR values—the results are displayed in Figure 7 with SNR=1 [Middle Panel] and SNR = 3[Right Panel].

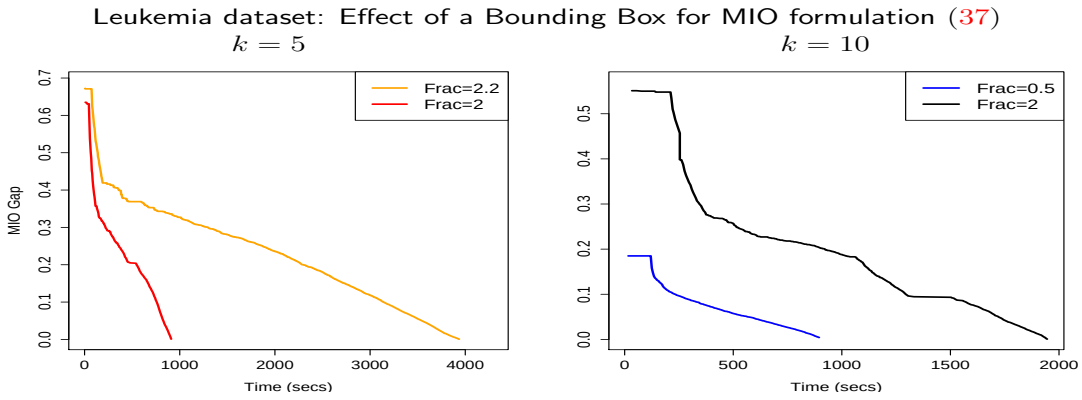

Figure 6: The effect of the MIO formulation (37) for the Leukemia dataset, for different values of k. Here $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$ and $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \text{Frac}$. For each value of k, the global minimum obtained was the same for the different choices of $\mathcal{L}^{\beta}_{\ell,\text{loc}}$.

In the same vein, we have Figure 12 studying the effect of formulations (37) for synthetic datasets generated as per Example 1 with $n = 50, p = 1000, \rho = 0.9$ and $k_0 = 5$.

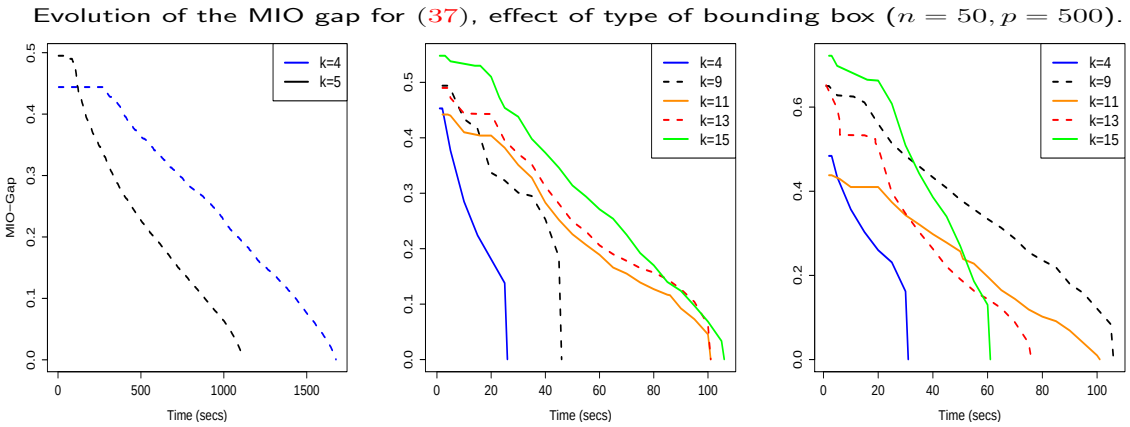

Figure 7: The effect of the MIO formulation (37) for a synthetic dataset as in Example 1 with $\rho = 0.9$, $k_0 = 5$, $n = 50, p = 500$, for different values of k. [Left Panel] $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = 0.5\|\mathbf{X}\boldsymbol{\beta}_0\|_1$ and $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \infty$ for a data-set with SNR $= 3$. [Middle Panel] $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$, $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \|\boldsymbol{\beta}_0\|_1/k$ and SNR $= 1$. [Right Panel] $\mathcal{L}^{\zeta}_{\ell,\text{loc}} = \infty$, $\mathcal{L}^{\beta}_{\ell,\text{loc}} = \|\boldsymbol{\beta}_0\|_1/k$ and SNR $= 3$. The figure shows that the bounding boxes in terms of $\mathbf{X}\boldsymbol{\beta}$ (left-panel) make the problem harder to solve, when compared to bounding boxes around $\boldsymbol{\beta}$ (middle and right panels). A possible reason is due to the strong correlations among the columns of $\mathbf{X}$. The SNR values do not seem to have a big impact on the run-times of the algorithms (middle and right panels).

5.3.3 Statistical Performance

To understand the statistical behavior of MIO when compared to other approaches for learning sparse models, we considered synthetic datasets for values of n ranging from $30-50$ and values of p ranging from $1000-2000$. The following methods were used for comparison purposes **(a)** Algorithm 2. Here we used fifty different random initializations around $\mathbf{0}$, of the form $\min(i-1,1)N(\mathbf{0}_{p\times 1}, 4\mathbf{I}), i=1,\ldots,50$ and took the solution corresponding to the best objective value; **(b)** The MIO approach with warm starts from part (a); **(c)** The `Lasso` solution and **(d)** The `Sparsenet` solution.

For methods (a), (b) we considered ten equi-spaced values of k in the range $[3, 2k_0]$ (including the optimal value of k_0). For each of the methods, the best model was selected in the same fashion as described in Section 5.2.3 using separate validation sets.

In addition, for some examples, we also study the performance of the *debiased* version of the `Lasso`, as described in Section 5.2.3.

In Figure 8 and Figure 9 we present selected representative results from four different examples described in Section 5.1.

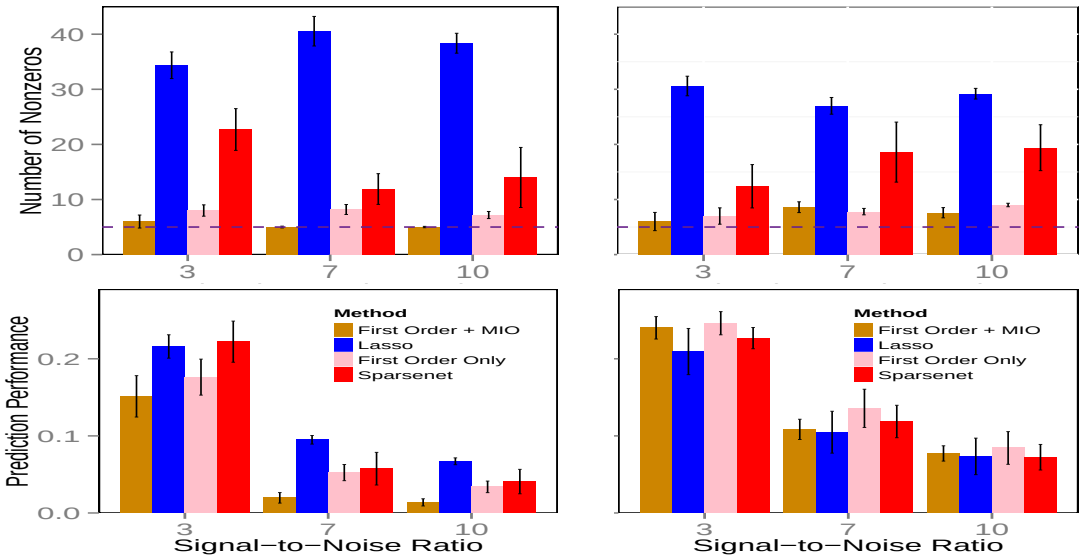

Figure 8: The sparsity and predictive performance for different procedures: [Left Panel] shows Example 1 with $n=50, p=1000, \rho=0.8, k_0=5$ and [Right Panel] shows Example 2 with $n=30, p=1000$—for each instance several SNR values have been shown.

In Figure 8 the left panel shows the performance of different methods for Example 1 with $n=50, p=1000, \rho=0.8, k_0=5$. In this example, there are five non-zero

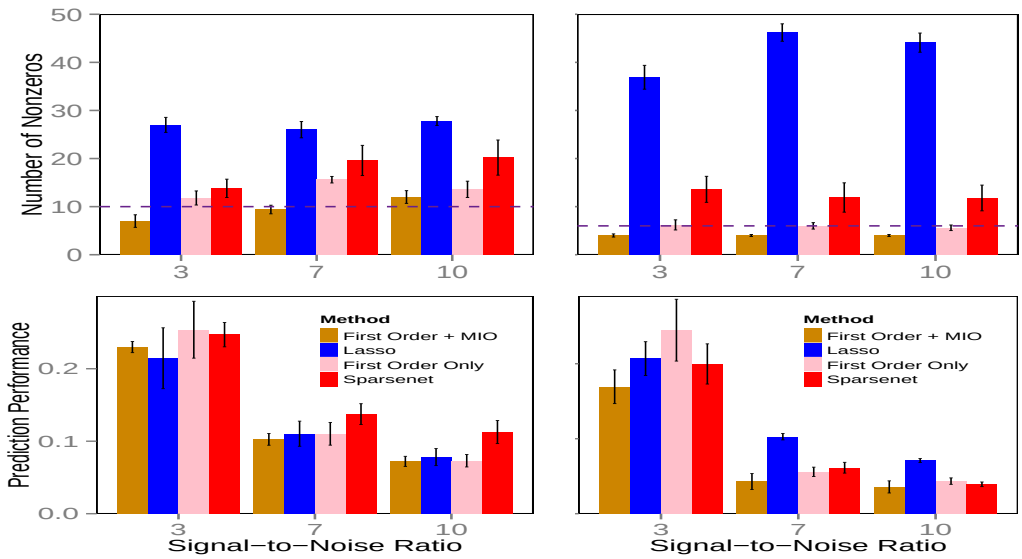

Figure 9: [Left Panel] Shows performance for data generated according to Example 3 with $n = 30, p = 1000$ and [Right Panel] shows Example 4 with $n = 50, p = 2000$.

coefficients: the features corresponding to the non-zero coefficients are weakly correlated and a feature having a non-zero coefficient is highly correlated with a feature having a zero coefficient. In this situation, the `Lasso` selects a very dense model since it fails to distinguish between a zero and a non-zero coefficient when the variables are correlated—it brings both the coefficients in the model (with shrinkage). MIO (with warm-start) performs the best—both in terms of predictive accuracy and in selecting a sparse set of coefficients. MIO obtains the sparsest model among the four methods and seems to find better solutions in terms of statistical properties than the models obtained by the first order methods alone. Interestingly, the "optimal model" selected by the first order methods is more dense than that selected by the MIO. The number of non-zero coefficients selected by MIO remains fairly stable across different SNR values, unlike the other three methods. For this example, we also experimented with the different versions of debiased `Lasso`. In summary: the best debiased `Lasso` models had performance marginally better than `Lasso` but quite inferior to MIO. See the results in Appendix, Section D.2 for further details.

In Figure 8 the right panel shows Example 2, with $n = 30, p = 1000, k_0 = 5$ and all non-zero coefficients equal one. In this example, all the methods perform similarly in terms of predictive accuracy. This is because all non-zero coefficients in $\boldsymbol{\beta}^0$ have the same value. In fact for the smallest value of SNR, the `Lasso` achieves the best predictive model. In all the cases however, the MIO achieves the sparsest model with favorable predictive accuracy.

In Figure 9, for both the examples, the model matrix is an iid Gaussian ensemble. The underlying regression coefficient $\boldsymbol{\beta}^0$ however, is structurally different than Example 2 (as in Figure 8, right-panel). The structure in $\boldsymbol{\beta}^0$ is responsible for different statistical behaviors of the four methods across Figures 8 (right-panel) and Figure 9 (both panels). The alternating signs and varying amplitudes of $\boldsymbol{\beta}^0$ are responsible for the poor behavior of `Lasso`. The MIO (with warm-starts) seems to be the best among all the methods. For Example 3 (Figure 9, left panel) the predictive performances of `Lasso` and MIO are comparable—the MIO however delivers much sparser models than the `Lasso`.

The key conclusions are as follows:

1. The MIO best subset algorithm has a significant edge in detecting the correct sparsity structure for all examples compared to `Lasso`, `Sparsenet` and the stand-alone discrete first order method.
2. For data generated as per Example 1 with large values of ρ, the MIO best subset algorithm gives better predictive performance compared to its competitors.
3. For data generated as per Examples 2 and 3, MIO delivers similar predictive models like the `Lasso`, but produces much sparser models. In fact, `Lasso` seems to perform marginally better than MIO, as a predictive model for small values of SNR.
4. For Example 4, MIO performs the best both in terms of predictive accuracy and delivering sparse models.

6 Computational Results for Subset Selection with Least Absolute Deviation Loss

In this section, we demonstrate how our method can be used for the best subset selection problem with LAD objective (34).

Since the main focus of this paper is the least squares loss function, we consider only a few representative examples for the LAD case. The LAD loss is appropriate when the error follows a heavy tailed distribution. The datasets used for the experiments parallel those described in Section 5.1, the difference being in the distribution of ϵ. We took ϵ_i iid from a double exponential distribution with variance σ^2. The value of σ^2 was adjusted to get different values of SNR.

Datasets analysed We consider a set-up similar to Example 1 (Section 5.1) with $k_0 = 5$ and $\rho = 0.9$. Different choices of (n, p) were taken to cover both the overdetermined

$(n = 500, p = 100)$ and high-dimensional cases $(n = 50, p = 1000$ and $n = 500, p = 1000)$.

The other competing methods used for comparison were (a) discrete first order method (Section (3.4)) (b) MIO warm-started with the first order solutions and (c) the LAD loss with ℓ_1 regularization:

$$\min \quad \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|_1,$$

which we denote by `LAD-Lasso`. The training, validation and testing were done in the same fashion as in the least squares case. For each method, we report the number of non-zeros in the optimal model and associated prediction accuracy (36).

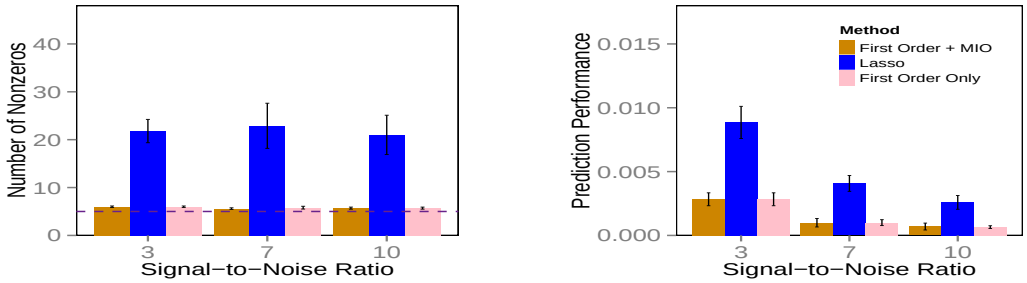

Figure 10: The sparsity and predictive performance for different procedures for $n = 500, p = 100$ for Problem (34). The data is generated as per Example 1 with $\rho = 0.9, k_0 = 5$ and double exponential errors—further details are available in the text. The acronym "Lasso" refers to `LAD-Lasso` (6). The MIO is seen to deliver sparser models with better predictive accuracy when compared to the `LAD-Lasso`.

Figure 10 compares the MIO approach with others for LAD in the overdetermined case $(n > p)$. Figure 11 does the same for the high-dimensional case $(p \gg n)$. The conclusions parallel those for the least squares case. Since, in the example considered, the features corresponding to the non-zero coefficients are weakly correlated and a feature having a non-zero coefficient is highly correlated with a feature having a zero coefficient—the `LAD-Lasso` selects an overly dense model and misses out in terms of prediction error. Both the MIO (with warm-starts) and the discrete first order methods behave similarly—much better than ℓ_1 regularization schemes. As expected, we observed that subset selection with least squares loss leads to inferior models for these examples, due to a heavy-tailed distribution of the errors.

The results in this section are similar to the least squares case. The MIO approach provides an edge both in terms of sparsity and predictive accuracy compared to Lasso both for the overdetermined and the high-dimensional case.

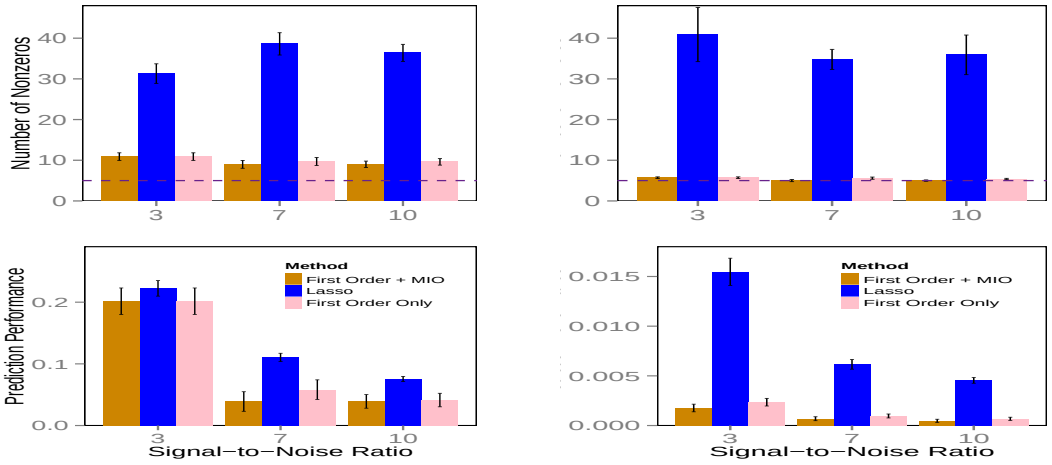

Figure 11: Figure showing the number of nonzero values and predictive performance for different values of n and p for Problem (34) (as in Figure 10). [Left panel] has $n = 50, p = 1000$ and [Right panel] has $n = 500, p = 1000$.

7 Conclusions

In this paper, we have revisited the classical best subset selection problem of choosing k out of p features in linear regression given n observations using a modern optimization lens, i.e., MIO and a discrete extension of first order methods from continuous optimization. Exploiting the astonishing progress of MIO solvers in the last twenty-five years, we have shown that this approach solves problems with n in the 1000s and p in the 100s in minutes to provable optimality, and finds near optimal solutions for n in the 100s and p in the 1000s in minutes. Importantly, the solutions provided by the MIO approach significantly outperform other state of the art methods like `Lasso` in achieving sparse models with good predictive power. Unlike all other methods, the MIO approach always provides a guarantee on its sub-optimality even if the algorithm is terminated early. Moreover, it can accommodate side constraints on the coefficients of the linear regression and also extends to finding best subset solutions for the least absolute deviation loss function.

While continuous optimization methods have played and continue to play an important role in statistics over the years, discrete optimization methods have not. The evidence in this paper as well as in [2] suggests that MIO methods are tractable and lead to desirable properties (improved accuracy and sparsity among others) at the expense of higher, but still reasonable, computational times.

Acknowledgements

We would like to thank the Associate editor and two reviewers for their comments that helped us improve the paper. A major part of the work was performed when R.M. was at Columbia University.

References

[1] Top500 Supercomputer Sites, Directory page for Top500 lists. Result for each list since June 1993. http://www.top500.org/statistics/sublist/. Accessed: 2013-12-04.

[2] D. Bertsimas and R. Mazumder. Least quantile regression via modern optimization. *The Annals of Statistics*, 42(6):2494–2525, 2014.

[3] D. Bertsimas and R. Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22, 2009.

[4] D. Bertsimas and R. Weismantel. *Optimization over integers*. Dynamic Ideas Belmont, 2005.

[5] P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.

[6] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming*, 74(2):121–140, 1996.

[7] R. E. Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica, Extra Volume: Optimization Stories*, pages 107–121, 2012.

[8] T. Blumensath and M. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[9] T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[11] P. Bühlmann and S. van-de-Geer. *Statistics for high-dimensional data*. Springer, 2011.

[12] F. Bunea, A. B. Tsybakov, M. H. Wegkamp, et al. Aggregation for gaussian regression. *The Annals of Statistics*, 35(4):1674–1697, 2007.

[13] E. Candes, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[14] E. Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[15] E. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2009.

[16] E. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52 (12):5406–5425, 2006.

[17] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[18] M. Dettling. Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593, 2004.

[19] D. Donoho. For most large underdetermined systems of equations, the minimal ℓ^1-norm solution is the sparsest solution. *Communications on Pure and Applied Mathematics*, 59:797–829, 2006.

[20] D. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1993.

[21] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[22] D. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *Information Theory, IEEE Transactions on*, 47(7):2845–2862, 2001.

[23] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression (with discussion). *Annals of Statistics*, 32(2):407–499, 2004. ISSN 0090-5364.

[24] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360(13), 2001.

[25] J. Fan and J. Lv. Nonconcave penalized likelihood with NP-dimensionality. *Information Theory, IEEE Transactions on*, 57(8):5467–5484, 2011.

[26] Y. Fan and J. Lv. Asymptotic equivalence of regularization methods in thresholded parameter space. *Journal of the American Statistical Association*, 108(503):1044–1061, 2013.

[27] I. Frank and J. Friedman. A statistical view of some chemometrics regression tools (with discussion). *Technometrics*, 35(2):109–148, 1993.

[28] J. Friedman. Fast sparse regression and classification. Technical report, Department of Statistics, Stanford University, 2008.

[29] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 2(1):302–332, 2007.

[30] G. Furnival and R. Wilson. Regression by leaps and bounds. *Technometrics*, 16: 499–511, 1974.

[31] E. Greenshtein. Best subset selection, persistence in high-dimensional statistical learning and optimization under ℓ_1 constraint. *The Annals of Statistics*, 34(5): 2367–2386, 2006.

[32] E. Greenshtein and Y. Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 2004.

[33] I. Gurobi Optimization. Gurobi optimizer reference manual, 2013. URL http://www.gurobi.com.

[34] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction (Springer Series in Statistics)*. Springer New York, 2 edition, 2009. ISBN 0387848576.

[35] K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28(5):1356–1378, 2000.

[36] P.-L. Loh and M. Wainwright. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.

[37] J. Lv and Y. Fan. A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, pages 3498–3528, 2009.

[38] R. Mazumder, J. Friedman, and T. Hastie. Sparsenet: Coordinate descent with non-convex penalties. *Journal of the American Statistical Association*, 117(495): 1125–1138, 2011.

[39] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[40] A. Miller. *Subset selection in regression*. CRC Press Washington, 2002.

[41] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[42] G. Nemhauser. Integer programming: the global impact. Presented at EURO, INFORMS, Rome, Italy, 2013. http://euro2013.org/wp-content/uploads/Nemhauser_EuroXXVI.pdf. Accessed: 2013-12-04.

[43] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming, Series A*, 103:127–152, 2005.

[44] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007. Technical Report number 76.

[45] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Norwell, 2004.

[46] G. Raskutti, M. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over-balls. *Information Theory, IEEE Transactions on*, 57(10):6976–6994, 2011.

[47] R. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1996. ISBN 0691015864. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0691015864.

[48] X. Shen, W. Pan, Y. Zhu, and H. Zhou. On constrained and regularized high-dimensional regression. *Annals of the Institute of Statistical Mathematics*, 65(5): 807–832, 2013.

[49] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[50] R. Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3): 273–282, 2011.

[51] J. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *Information Theory, IEEE Transactions on*, 52(3):1030–1051, 2006.

[52] S. 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.

[53] M. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5):2183–2202, 2009.

[54] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[55] C.-H. Zhang and J. Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *Annals of Statistics*, 36(4):1567–1594, 2008.

[56] C.-H. Zhang and T. Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

[57] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107, 2010.

[58] Y. Zhang, M. Wainwright, and M. I. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. *arXiv preprint arXiv:1402.1918*, 2014.

[59] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[60] Z. Zheng, Y. Fan, and J. Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014. ISSN 1467-9868.

[61] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

[62] H. Zou and R. Li. One - step sparse estimates in nonconcave penalized likelihood problems. *The Annals of Statistics*, 36(4):1509–1533, 2008.

Appendix and Supplementary Material

A Additional Details for Section 2

A.1 Solving the convex quadratic optimization Problems in Section 2.3.2

We show here that the convex quadratic optimization problems appearing in Section 2.3.2 are indeed quite simple and can be solved with small computational cost.

We first consider Problem (18), the computation of u_i^- which is a minimization problem. We assume without loss of generality that the feasible set of problem (18) is non-empty. Thus by standard results in quadratic optimization [10], it follows that, there exists a τ such that:

$$\nabla \left(\frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \tau \beta_i \right) = 0,$$

where, ∇ denotes derivative wrt $\boldsymbol{\beta}$ and a $\boldsymbol{\beta}$ that satisfies the above gradient condition must also be feasible for Problem (18). Simplifying the above equation, we get:

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{y} - \tau e_i,$$

where, e_i is a vector in $\Re^p$ such that its ith coordinate is one with the remaining equal to zero. Simplifying the above expression, we have

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \|(\mathbb{I} - P_X)\mathbf{y} + \tau q_i\|_2^2.$$

Above, $\mathbb{I}$ is the identity matrix of size $p \times p$ and P_X is the familiar projection matrix given by $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ [3] and $q_i = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}e_i$. Observing that $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \text{UB}$, one can readily estimate τ that satisfies the above simple quadratic equation. This leads to the solution of τ, which subsequently leads to the optimal value $\widetilde{\boldsymbol{\beta}}$ that solves Problem (18). This readily leads to the optimum of Problem (18).

The above argument readily applies to Problem (18), for the computation of u_i^+ by writing it as an equivalent minimization problem and observing that:

$$-u_i^+ = \min_{\boldsymbol{\beta}} \quad -\beta_i \quad s.t. \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \leq \text{UB}.$$

The above derivation can also be adapted to the case of Problem (19). Towards this end, notice that for estimating v_i^- the above steps (for computing u_i^-) will be modified:

[3] Note that we assume here that $p > n$ which typically guarantees that $\mathbf{X}'\mathbf{X}$ is invertible with probability one, provided the entries of $\mathbf{X}$ are drawn from a continuous ditribution.

e_i gets replaced by $\mathbf{x}_i \in \Re^p$ (the ith row of $\mathbf{X}$); and P_X denotes the projection matrix onto the column space of $\mathbf{X}$, even if the matrix $\mathbf{X}'\mathbf{X}$ is not invertible (since here, we consider arbitrary n, p).

In addition, the Problems (18) for the different variables and (19) for the different samples; can be solved *completely* independently, in parallel.

A.2 Details for Section 2.3.4

Note that in Problem (10) we consider a uniform bound on β_i's: $-\mathcal{M}_U \leq \beta_i \leq \mathcal{M}_U$, for all $i = 1, \ldots, p$. Note that some of the variables β_i may have larger amplitude than the others, thus it may be reasonable to have bounds depending upon the variable index i. Thusly motivated, for added flexibility, one can consider the following (adaptive) bounds on β_i's: $-\mathcal{M}_U^i \leq \beta_i \leq \mathcal{M}_U^i$ for $i = 1, \ldots, p$. The parameters $\mathcal{M}_U^i$ can be taken as $\max\{|u_i^+|, |u_i^-|\}$, as defined in (18).

More generally, one can also consider asymmetric bounds on β_i as: $u_i^- \leq \beta_i \leq u_i^+$ for all i.

Note that the above ideas for bounding β_i's can also be extended to obtain sample-specific bounds on $\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle$ for $i = 1, \ldots, n$.

The bounds on $\|\widehat{\boldsymbol{\beta}}\|_1$ and $\|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_1, \|\mathbf{X}\widehat{\boldsymbol{\beta}}\|_\infty$ can also be adapted to the above variable dependent bounds on β_i's.

While the above modifications may lead to marginally improved performances, we do not dwell much on these improvements mainly for the sake of a clear exposition.

A.3 Proof of Proposition 2

Proof

(a) Given a set I, we define $\mathbf{G} := \mathbf{X}_I'\mathbf{X}_I - \mathbf{I}$, and let g_{ij} denote the (i, j)th entry of

$\mathbf{G}$. For any $\mathbf{u} \in \mathbb{R}^k$ we have

$$\begin{aligned}
\max_{\|\mathbf{u}\|_1=1} \|\mathbf{G}\mathbf{u}\|_1 &= \max_{\|\mathbf{u}\|_1=1} \left(\sum_{i=1}^{k} \left| \sum_{j=1}^{k} g_{ij} u_j \right| \right) \\
&\leq \max_{\|\mathbf{u}\|_1=1} \left(\sum_{i=1}^{k} \sum_{j=1}^{k} |u_j||g_{ij}| \right) \\
&= \max_{\|\mathbf{u}\|_1=1} \left(\sum_{j=1}^{k} |u_j| \sum_{i \neq j} |g_{ij}| \right) \qquad (g_{jj} = 0) \\
&\leq \max_{\|\mathbf{u}\|_1=1} (\mu[k-1]\|\mathbf{u}\|_1) \qquad \left(\sum_{i \neq j} |g_{ij}| \leq \mu[k-1] \right) \\
&= \mu[k-1].
\end{aligned}$$

(b) Using $\mathbf{X}_I'\mathbf{X}_I = \mathbf{I} + \mathbf{G}$ and standard power-series convergence (which is valid since $\|\mathbf{G}\|_{1,1} < 1$) we obtain

$$\|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{1,1} = \| \left(\mathbf{I} + \mathbf{G}\right)^{-1} \|_{1,1} = \sum_{i=0}^{\infty} \|\mathbf{G}\|_{1,1}^i \leq \frac{1}{1 - \|\mathbf{G}\|_{1,1}} \leq \frac{1}{1 - \mu[k-1]}. \qquad \square$$

A.4 Proof of Theorem 2.1

Proof

(a) Since $\widehat{\boldsymbol{\beta}}_I = (\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'\mathbf{y}$ we have

$$\|\widehat{\boldsymbol{\beta}}\|_1 = \|\widehat{\boldsymbol{\beta}}_I\|_1 \leq \|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\|_{1,1} \|\mathbf{X}_I'\mathbf{y}\|_1. \tag{38}$$

Note that

$$\|\mathbf{X}_I'\mathbf{y}\|_1 = \sum_{j \in I} |\langle \mathbf{X}_j, \mathbf{y} \rangle| \leq \max_{I, |I|=k} \sum_{j \in I} |\langle \mathbf{X}_j, \mathbf{y} \rangle| \leq \sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y} \rangle|. \tag{39}$$

Applying Part (b) of Proposition 2 and (39) to (38), we obtain (14) .

(b) We write $\widehat{\boldsymbol{\beta}}_I = \mathbf{A}\mathbf{y}$ for $\mathbf{A} = (\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'$. If $\mathbf{a}_i, i = 1, \ldots, k$ denote the rows of $\mathbf{A}$ we have:

$$\|\widehat{\boldsymbol{\beta}}_I\|_\infty = \max_{i=1,\ldots,k} |\langle \mathbf{a}_i, \mathbf{y} \rangle| \leq \left(\max_{i=1,\ldots,k} \|\mathbf{a}_i\|_2 \right) \|\mathbf{y}\|_2. \tag{40}$$

For every $i = 1, \ldots, k$ we have

$$\begin{aligned}\|\mathbf{a}_i\|_2 \leq & \max_{\|\mathbf{u}\|_2=1} \|\mathbf{A}\mathbf{u}\|_2 \\ = & \max_{\|\mathbf{u}\|_2=1} \|(\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'\mathbf{u}\|_2 \\ \leq & \lambda_{\max}\left((\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'\right) \\ = & \max\left\{\frac{1}{d_1}, \ldots, \frac{1}{d_k}\right\}, \end{aligned} \tag{41}$$

where $d_1, \ldots, d_k$ are the (nonzero) singular values of the matrix $\mathbf{X}_I$. To see how one arrives at (41) let us denote the singular value decomposition of $\mathbf{X}_I = \mathbf{U}\mathbf{D}\mathbf{V}'$ with $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_k)$. We then have

$$(\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I' = (\mathbf{V}\mathbf{D}^{-2}\mathbf{V}')(\mathbf{U}\mathbf{D}\mathbf{V}')' = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}'$$

and the singular values of $(\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'$ are thus $1/d_i$, $i = 1, \ldots, k$.

The eigenvalues of $\mathbf{X}_I'\mathbf{X}_I$ are d_i^2 and from (12) we obtain that $d_i^2 \geq \eta_k$. Using (41) we thus obtain

$$\max_{i=1,\ldots,k} \|\mathbf{a}_i\|_2 \leq \frac{1}{\sqrt{\eta_k}}. \tag{42}$$

Substituting the bound (42) to (40) we obtain

$$\|\widehat{\boldsymbol{\beta}}_I\|_\infty \leq \frac{1}{\sqrt{\eta_k}}\|\mathbf{y}\|_2. \tag{43}$$

Using the notation $\tilde{\mathbf{A}} = (\mathbf{X}_I'\mathbf{X}_I)^{-1}$, we have

$$\begin{aligned}\|\widehat{\boldsymbol{\beta}}_I\|_\infty = & \max_{i=1,\ldots,k} |\langle \tilde{\mathbf{a}}_i, \mathbf{X}_I'\mathbf{y}\rangle| \\ \leq & \left(\max_{i=1,\ldots,k|} \|\tilde{\mathbf{a}}_i\|_2\right) \|\mathbf{X}_I'\mathbf{y}\|_2 \\ \leq & \lambda_{\max}\left((\mathbf{X}_I'\mathbf{X}_I)^{-1}\right) \|\mathbf{X}_I'\mathbf{y}\|_2 \\ = & \left(\max_{i=1,\ldots,k} \frac{1}{d_i^2}\right) \cdot \sqrt{\sum_{j\in I} |\langle \mathbf{X}_j, \mathbf{y}\rangle|^2} \\ \leq & \frac{1}{\eta_k}\sqrt{\sum_{j=1}^{k} |\langle \mathbf{X}_{(j)}, \mathbf{y}\rangle|^2}. \end{aligned} \tag{44}$$

Combining (43) and (44) we obtain (15).

(c) We have

$$\|\mathbf{X}_I\widehat{\boldsymbol{\beta}}_I\|_1 \le \sum_{i=1}^n |\langle \mathbf{x}_i, \widehat{\boldsymbol{\beta}}_I\rangle| \le \sum_{i=1}^n \|\mathbf{x}_i\|_\infty \|\widehat{\boldsymbol{\beta}}_I\|_1 = \sum_{i=1}^n \|\mathbf{x}_i\|_\infty \|\widehat{\boldsymbol{\beta}}_I\|_1. \tag{45}$$

Let $\mathbf{P}_I := \mathbf{X}_I(\mathbf{X}_I'\mathbf{X}_I)^{-1}\mathbf{X}_I'$ denote the projection onto the columns of $\mathbf{X}_I$. We have $\|\mathbf{P}_I\mathbf{y}\|_2 \le \|\mathbf{y}\|_2$, leading to:

$$\|\mathbf{X}_I\widehat{\boldsymbol{\beta}}_I\|_1 = \|\mathbf{P}_I\mathbf{y}\|_1 \le \sqrt{k}\|\mathbf{P}_I\mathbf{y}\|_2 \le \sqrt{k}\|\mathbf{y}\|_2, \tag{46}$$

where we used that for any $\mathbf{a} \in \mathbb{R}^m$, we have $\sqrt{m}\|\mathbf{a}\|_2 \ge \|\mathbf{a}\|_1$. Combining (45) and (46) we obtain (16).

(d) For any vector $\boldsymbol{\beta}_I$ which has zero entries in the coordinates outside I, we have:

$$\|\mathbf{X}\boldsymbol{\beta}_I\|_\infty \le \max_{i=1,\dots,n} |\langle \mathbf{x}_i, \boldsymbol{\beta}_I\rangle| \le \max_{i=1,\dots,n} \|\mathbf{x}_i\|_{1:k}\|\boldsymbol{\beta}_I\|_\infty,$$

leading to (17). □

B Proofs and Technical Details for Section 3

B.1 Proof of Proposition 6

Proof

(a) Let $\boldsymbol{\beta}$ be a vector satisfying $\|\boldsymbol{\beta}\|_0 \le k$. Using the notation $\widehat{\boldsymbol{\eta}} \in \mathbf{H}_k\left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right)$ we have the following chain of inequalities:

$$\begin{aligned}
g(\boldsymbol{\beta}) &= Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}) \\
&\ge \inf_{\|\boldsymbol{\eta}\|_0 \le k} \; Q_L(\boldsymbol{\eta}, \boldsymbol{\beta}) \\
&= \inf_{\|\boldsymbol{\eta}\|_0 \le k} \; \left(\frac{L}{2}\|\boldsymbol{\eta} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \boldsymbol{\eta} - \boldsymbol{\beta}\rangle + g(\boldsymbol{\beta})\right) \\
&= \inf_{\|\boldsymbol{\eta}\|_0 \le k} \; \left(\frac{L}{2}\left\|\boldsymbol{\eta} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right)\right\|_2^2 - \frac{1}{2L}\|\nabla g(\boldsymbol{\beta})\|_2^2 + g(\boldsymbol{\beta})\right) \\
&= \left(\frac{L}{2}\|\widehat{\boldsymbol{\eta}} - \left(\boldsymbol{\beta} - \frac{1}{L}\nabla g(\boldsymbol{\beta})\right)\|_2^2 - \frac{1}{2L}\|\nabla g(\boldsymbol{\beta})\|_2^2 + g(\boldsymbol{\beta})\right) \qquad \text{(From (26))} \\
&= \left(\frac{L}{2}\|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\rangle + g(\boldsymbol{\beta})\right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{L-\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + \frac{\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \widehat{\boldsymbol{\eta}} - \boldsymbol{\beta} \rangle + g(\boldsymbol{\beta}) \right) \\
&\geq \frac{L-\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + \underbrace{\left(\frac{\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + \langle \nabla g(\boldsymbol{\beta}), \widehat{\boldsymbol{\eta}} - \boldsymbol{\beta} \rangle + g(\boldsymbol{\beta}) \right)}_{Q_\ell(\widehat{\boldsymbol{\eta}}, \boldsymbol{\beta})} \\
&\geq \frac{L-\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 + g(\widehat{\boldsymbol{\eta}}). \qquad \text{(From (25))}
\end{aligned}
$$

This chain of inequalities leads to:

$$
g(\boldsymbol{\beta}) - g(\widehat{\boldsymbol{\eta}}) \geq \frac{L-\ell}{2} \|\widehat{\boldsymbol{\eta}} - \boldsymbol{\beta}\|_2^2 . \tag{47}
$$

Applying (47) for $\boldsymbol{\beta} = \boldsymbol{\beta}_m$ and $\widehat{\boldsymbol{\eta}} = \boldsymbol{\beta}_{m+1}$, the vectors generated by Algorithm 1, we obtain (31). This implies that the objective values $g(\boldsymbol{\beta}_m)$ are decreasing and since the sequence is bounded below ($g(\boldsymbol{\beta}) \geq 0$), we obtain that $g(\boldsymbol{\beta}_m)$ converges as $m \to \infty$.

(b) If $L > \ell$ and from part (a), the result follows.

(c) The condition $\underline{\alpha}_k > 0$ means that for all m sufficiently large, the entry $|\beta_{(k),m}|$ will remain (uniformly) bounded away from zero. We will use this to prove that the support of $\boldsymbol{\beta}_m$ converges. For the purpose of establishing contradiction suppose that the support does not converge. Then, there are infinitely many values of m' such that $\mathbf{1}_{m'} \neq \mathbf{1}_{m'+1}$. Using the fact that $\|\boldsymbol{\beta}_m\|_0 = k$ for all large m we have

$$
\|\boldsymbol{\beta}_{m'} - \boldsymbol{\beta}_{m'+1}\|_2 \geq \sqrt{\beta_{m',i}^2 + \beta_{m'+1,j}^2} \geq \frac{|\beta_{m',i}| + |\beta_{m'+1,j}|}{\sqrt{2}}, \tag{48}
$$

where i, j are such that $\beta_{m'+1,i} = \beta_{m',j} = 0$. As $m' \to \infty$, the quantity in the rhs of (48) remains bounded away from zero since $\underline{\alpha}_k > 0$. This contradicts the fact that $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to \mathbf{0}$, as established in part (b). Thus, $\mathbf{1}_m$ converges, and since $\mathbf{1}_m$ is a discrete sequence, it converges after finitely many iterations, that is $\mathbf{1}_m = \mathbf{1}_{m+1}$ for all $m \geq M^*$. Algorithm 1 becomes a vanilla gradient descent algorithm, restricted to the space $\mathbf{1}_m$ for $m \geq M^*$. Since a gradient descent algorithm for minimizing a convex function over a closed convex set leads to a sequence of iterates that converge [47, 45], we conclude that Algorithm 1 converges. Therefore, the sequence $\boldsymbol{\beta}_m$ converges to $\boldsymbol{\beta}^*$, a first order stationarity point:

$$
\boldsymbol{\beta}^* \in \mathbf{H}_k \left(\boldsymbol{\beta}^* - \frac{1}{L} \nabla g(\boldsymbol{\beta}^*) \right).
$$

(d) Let $\mathcal{I}_m \subset \{1,\ldots,p\}$ denote the set of k largest values of the vector $\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)$ in absolute value. By the definition of $\mathbf{H}_k\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)$, we have

$$\left|\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)_i\right| \geq \left|\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)_j\right|,$$

for all i, j with $i \in \mathcal{I}_m$ and $j \notin \mathcal{I}_m$. Thus,

$$\liminf_{m\to\infty} \min_{i\in\mathcal{I}_m} \left|\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)_i\right| \geq \liminf_{m\to\infty} \max_{j\notin\mathcal{I}_m} \left|\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)_j\right|. \tag{49}$$

Moreover,

$$\left(\boldsymbol{\beta}_m - \mathbf{H}_k\left(\boldsymbol{\beta}_m - \frac{1}{L}\nabla g(\boldsymbol{\beta}_m)\right)\right)_i = \begin{cases} \frac{1}{L}(\nabla g(\boldsymbol{\beta}_m))_i, & i \in \mathcal{I}_m, \\ \beta_{m,i}, & \text{otherwise.} \end{cases}$$

Using the fact that $\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m \to \mathbf{0}$ we have

$$(\nabla g(\boldsymbol{\beta}_m))_i \to 0, i \in \mathcal{I}_m \text{ and } \beta_{m,j} \to 0, j \notin \mathcal{I}_m$$

as $m \to \infty$. Combining with (49) we have that:

$$\liminf_{m\to\infty} \min_{i\in\mathcal{I}_m} |\boldsymbol{\beta}_{mi}| \geq \liminf_{m\to\infty} \max_{j\notin\mathcal{I}_m} \frac{1}{L}\left|(\nabla g(\boldsymbol{\beta}_m))_j\right| = \frac{1}{L}\liminf_{m\to\infty} \|\nabla g(\boldsymbol{\beta}_m)\|_\infty.$$

Since, $\liminf_{m\to\infty} \min_{i\in\mathcal{I}_m} |\boldsymbol{\beta}_{mi}| = \underline{\alpha}_k = 0$ (by hypothesis), the lhs of the above inequality equals zero, which leads to $\liminf_{m\to\infty} \|\nabla g(\boldsymbol{\beta}_m)\|_\infty = 0$.

(e) We build on the proof of Part (d).

It follows from equation (49) (by suitably modifying 'lim inf' to 'lim sup') that:

$$\underbrace{\limsup_{m\to\infty} \min_{i\in\mathcal{I}_m} |\boldsymbol{\beta}_{mi}|}_{\overline{\alpha}_k} \geq \limsup_{m\to\infty} \max_{j\notin\mathcal{I}_m} \frac{1}{L}\left|(\nabla g(\boldsymbol{\beta}_m))_j\right| = \frac{1}{L}\limsup_{m\to\infty} \|\nabla g(\boldsymbol{\beta}_m)\|_\infty.$$

Note that the lhs of the above inequality is $\overline{\alpha}_k$ which is zero (by hypothesis), thus $\|\nabla g(\boldsymbol{\beta}_m)\|_\infty \to 0$ as $m \to \infty$.

Suppose $\boldsymbol{\beta}_\infty$ is a limit point of the sequence $\boldsymbol{\beta}_m$. Thus there is a subsequence $m' \subset \{1, 2, \ldots,\}$ such that $\boldsymbol{\beta}_{m'} \to \boldsymbol{\beta}_\infty$ and $g(\boldsymbol{\beta}_{m'}) \to g(\boldsymbol{\beta}_\infty)$. Using the continuity of the gradient and hence the function $\cdot \mapsto \|\nabla g(\cdot)\|_\infty$ we have that $\|\nabla g(\boldsymbol{\beta}_{m'})\|_\infty \to \|\nabla g(\boldsymbol{\beta}_\infty)\|_\infty = 0$ as $m' \to \infty$. Thus $\boldsymbol{\beta}_\infty$ is a solution to the unconstrained (without cardinality constraints) optimization problem $\min\ g(\boldsymbol{\beta})$. Since $g(\boldsymbol{\beta}_m)$ is a decreasing sequence, $g(\boldsymbol{\beta}_m)$ converges to the minimum of $g(\boldsymbol{\beta})$. □

B.2 Proof of Proposition 3

Proof:
We provide a proof of Proposition 3, for the sake of completeness.

It suffices to consider $|c_i| > 0$ for all i. Let $\boldsymbol{\beta}$ be an optimal solution to Problem (22) and let $S := \{i : \beta_i \neq 0\}$. The objective function is given by $\sum_{i \notin S} |c_i|^2 + \sum_{i \in S} (\beta_i - c_i)^2$. Note that by selecting $\beta_i = c_i$ for $i \in S$, we can make the objective function $\sum_{i \notin S} |c_i|^2$. Thus, to minimize the objective function, S must correspond to the indices of the largest k values of $|c_i|, i \geq 1$. □

B.3 Proof of Proposition 7

Proof
This follows from Proposition 6, Part (a), which implies that:

$$g(\boldsymbol{\eta}) - g(\hat{\boldsymbol{\eta}}) \geq \frac{L - \ell}{2} \left\| \hat{\boldsymbol{\eta}} - \boldsymbol{\eta} \right\|_2^2,$$

for any $\hat{\boldsymbol{\eta}} \in \mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta}) \right)$. Now by the definition of $\mathbf{H}_k(\cdot)$, we have $g(\boldsymbol{\eta}) = g(\hat{\boldsymbol{\eta}})$ which along with $L > \ell$ implies that the rhs of the above inequality is zero: thus $\left\| \hat{\boldsymbol{\eta}} - \boldsymbol{\eta} \right\|_2 = 0$, i.e., $\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}$. Since the choice of $\hat{\boldsymbol{\eta}}$ was arbitrary, it follows that $\boldsymbol{\eta}$ is the only element in the set $\mathbf{H}_k \left(\boldsymbol{\eta} - \frac{1}{L} \nabla g(\boldsymbol{\eta}) \right)$. □

B.4 Proof of Proposition 8

Proof
The proof follows by noting that $\widehat{\boldsymbol{\beta}}$ is k-sparse along with Proposition 6, Part (a), which implies that:

$$g(\widehat{\boldsymbol{\beta}}) - g(\hat{\boldsymbol{\eta}}) \geq \frac{L - \ell}{2} \left\| \widehat{\boldsymbol{\beta}} - \hat{\boldsymbol{\eta}} \right\|_2^2,$$

for any $\hat{\boldsymbol{\eta}} \in \mathbf{H}_k \left(\widehat{\boldsymbol{\beta}} - \frac{1}{L} \nabla g(\widehat{\boldsymbol{\beta}}) \right)$. Now, by the definition of $\widehat{\boldsymbol{\beta}}$ we have $g(\widehat{\boldsymbol{\beta}}) = g(\hat{\boldsymbol{\eta}})$ which along with $L > \ell$ implies that the rhs of the above inequality is zero: thus $\widehat{\boldsymbol{\beta}}$ is a first order stationary point. □

B.5 Proof of Theorem 3.1

Proof
Summing inequalities (31) for $1 \leq m \leq M$. we obtain

$$\sum_{m=1}^{M} \big(g(\boldsymbol{\beta}_m) - g(\boldsymbol{\beta}_{m+1})\big) \geq \frac{L-\ell}{2} \sum_{m=1}^{M} \|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2, \tag{50}$$

leading to

$$g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}_{M+1}) \geq \frac{M(L-\ell)}{2} \min_{m=1,\ldots,M} \|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2.$$

Since the decreasing sequence $g(\boldsymbol{\beta}_{m+1})$ converges to $g(\boldsymbol{\beta}^*)$ by Proposition 6 we have:

$$\frac{g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}^*)}{M} \geq \frac{g(\boldsymbol{\beta}_1) - g(\boldsymbol{\beta}_{M+1})}{M} \geq \frac{(L-\ell)}{2} \min_{m=1,\ldots,M} \|\boldsymbol{\beta}_{m+1} - \boldsymbol{\beta}_m\|_2^2. \qquad \square$$

B.6 Proof of Proposition 5

Proof
If $\boldsymbol{\eta}$ is a first order stationary point with $\|\boldsymbol{\eta}\|_0 \leq k$, it follows from the argument following Definition 1, that there is a set $I \subset \{1, \ldots, p\}$ with $|I^c| = k$ such that $\nabla_i g(\boldsymbol{\eta}) = 0$ for all $i \notin I$ and $\eta_i = 0$ for all $i \notin I$. Let $\mu_i := \eta_i - \frac{1}{L}\nabla_i g(\boldsymbol{\eta})$ for $i = 1, \ldots, p$. Suppose I_k denotes the set of indices corresponding to the top k ordered values of $|\mu_i|$. Note that:

$$\mu_i = \eta_i, \quad i \in I_k \quad \text{and} \quad |\mu_j| = |\frac{1}{L}\nabla_j g(\boldsymbol{\eta})|, \quad j \notin I_k. \tag{51}$$

For $i \in I_k$ and $j \notin I_k$ we have $|\mu_i| \geq |\mu_j|$. This implies that $|\eta_i| \geq |\frac{1}{L}\nabla_j g(\boldsymbol{\eta})|$. Since $\boldsymbol{\eta} \in \mathbf{H}_k\left(\boldsymbol{\eta} - \frac{1}{L}\nabla g(\boldsymbol{\eta})\right)$ and $\|\boldsymbol{\eta}\|_0 < k$, it follows that $0 = \min_{i \in I_k} |\eta_i| = \min_{i \in I_k} |\mu_i|$. We thus have that $\nabla_j g(\boldsymbol{\eta}) = 0$ for all $j \notin I_k$. In addition, note that $\nabla_i g(\boldsymbol{\eta}) = 0$ for all $i \in I_k$. Thus it follows that $\nabla g(\boldsymbol{\eta}) = \mathbf{0}$ and hence $\boldsymbol{\eta} \in \arg\min_{\boldsymbol{\eta}}\ g(\boldsymbol{\eta})$. $\square$

C Brief Review of Statistical Properties for the subset selection problem

In this section, for the sake of completeness we briefly review some of the properties of solutions to Problem (1).

Suppose the linear model assumption is true, i.e., $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0 + \boldsymbol{\epsilon}$, with $\epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$. Let $\widehat{\boldsymbol{\beta}}$ denote a solution to (1). [46] showed that, with probability greater than $1 - \exp(-c_1 k \log(p/k))$, the worst case (over $\boldsymbol{\beta}^0$) predictive performance has the following upper bound:

$$\max_{\boldsymbol{\beta}^0 : \|\boldsymbol{\beta}^0\|_0 \leq k} \frac{1}{n} \left\| \mathbf{X}\boldsymbol{\beta}^0 - \mathbf{X}\widehat{\boldsymbol{\beta}} \right\|_2^2 \leq c_2 \sigma^2 \frac{k \log(p/k)}{n}, \tag{52}$$

where, c_1, c_2 are universal constants. Similar results also appear in [12, 58]. Interestingly, the upper bound (52) does *not* depend upon $\mathbf{X}$. Unless $p/k = O(1)$, the upper bound appearing in (52) is of the order $O(\sigma^2 \frac{k \log(p)}{n})$ where the constants are universal. In terms of the expected (worst case) predictive risk, an upper bound is given by [58]:

$$\max_{\boldsymbol{\beta}^0 : \|\boldsymbol{\beta}^0\|_0 \leq k} \frac{1}{n} \mathbb{E} \left(\left\| \mathbf{X}\boldsymbol{\beta}^0 - \mathbf{X}\widehat{\boldsymbol{\beta}} \right\|_2^2 \right) \lesssim \sigma^2 \frac{k \log(p)}{n}, \tag{53}$$

where, the symbol "$\lesssim$" means "$\leq$" upto some universal constants.

A natural question is how do the bounds for `Lasso`-based solutions compare with (53)? In a recent paper [58], the authors derive upper and lower bounds of the prediction performance of the thresholded version of the `Lasso` solution, which we present briefly. Suppose

$$\hat{\boldsymbol{\beta}}_{\ell_1} \in \arg\min_{\boldsymbol{\beta}} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_n \|\boldsymbol{\beta}\|_1$$

denotes a `Lasso` solution for $\lambda_n = 4\sigma\sqrt{\frac{\log p}{n}}$. Let $\hat{\boldsymbol{\beta}}_{\text{TL}}$ denote the thresholded version of the `Lasso` solution, which retains the top k entries of $\hat{\boldsymbol{\beta}}_{\ell_1}$ in absolute value and sets the remaining to zero. The bounds on the predictive performances of `Lasso` based solutions depend upon a restricted eigen-value type condition. Following [58], we define, for any subset $S \in \{1, 2, \ldots, p\}$, the quantity: $C(S) := \{\boldsymbol{\beta} | \|\boldsymbol{\beta}_{S^c}\|_1 \leq 2\|\boldsymbol{\beta}_S\|_1\}$, where, $\|\boldsymbol{\beta}_S\|_1 = \sum_{j \in S} |\beta_j|$ and $\|\boldsymbol{\beta}_{S^c}\|_1 = \sum_{j \in S^c} |\beta_j|$. We say that the matrix $\mathbf{X}$ satisfies a restricted eigen-value type condition with parameter $\gamma(X)$ if it satisfies the following:

$$\frac{1}{n}\|\mathbf{X}\boldsymbol{\beta}\|_2^2 \geq \gamma(\mathbf{X})\|\boldsymbol{\beta}\|_2^2 \qquad \text{for} \quad \boldsymbol{\beta} \in \cup_{S:|S|=k} C(S).$$

Note that $\gamma(\mathbf{X}) \leq 1$ and $\gamma(\mathbf{X})$ is also related to the so called compatibility condition [11]. In an insightful paper, [58] show that under such restricted eigenvalue type conditions the following holds:

$$\frac{\sigma^2}{\gamma(\mathbf{X}_{\text{bad}})^2} \frac{k^{1-\delta} \log(p)}{n} \lesssim \max_{\boldsymbol{\beta}^0 : \|\boldsymbol{\beta}^0\|_0 \leq k} \frac{1}{n} \mathbb{E} \left(\left\| \mathbf{X}\boldsymbol{\beta}^0 - \mathbf{X}\widehat{\boldsymbol{\beta}}_{\text{TL}} \right\|_2^2 \right) \lesssim \frac{\sigma^2}{\gamma(\mathbf{X})^2} \frac{k \log(p)}{n} \tag{54}$$

In particular, the lower bounds apply to *bad* design matrices $\mathbf{X}_{\text{bad}}$ for some arbitrarily small scalar $\delta > 0$. In fact [58] establish a result stronger than (54), where, $\widehat{\boldsymbol{\beta}}_{\text{TL}}$ can be

replaced by a k-sparse estimate delivered by a polynomial time method. The bounds displayed in (54) show that there is a significant *gap* between the predictive performance of subset selection procedures (see bound (53)) and `Lasso` based k-sparse solutions—the magnitude of the gap depends upon how small $\gamma(\mathbf{X})$ is. $\gamma(\mathbf{X})$ can be small if the pairwise correlations between the features of the model matrix is quite high. These results complement our experimental findings in Section 5.

An in-depth analysis of the properties of solutions to the Lagrangian version of Problem (1), namely, Problem (4) is presented in [56]. [46, 56] also analyze the errors in the regression coefficients: $\|\boldsymbol{\beta}^0 - \widehat{\boldsymbol{\beta}}\|_2$, under further minor assumptions on the model matrix $\mathbf{X}$. [56, 48] provide interesting theoretical analysis of the variable selection properties of (1) and (4), showing that subset selection procedures have superior variable selection properties over `Lasso` based methods.

In passing, we remark that [56] develop statistical properties of *inexact* solutions to Problem (4). This may serve as interesting theoretical support for *near global* solutions to Problem (1), where the certificates of sub-optimality are delivered by our MIO framework in terms of global lower bounds. A precise and thorough understanding of the statistical properties of sub-optimal solutions to Problem (1) is left for an interesting piece of future work.

D Additional Details on Experiments and Computations

D.1 Some additional figures related to the radii of bounding boxes

Some figures illustrating the effect of the bounding box radii are presented in Figure 12.

D.2 `Lasso`, Debiased `Lasso` and MIO

We present here comparisons of the debiased `Lasso` with MIO and `Lasso`.

Debiasing is often used to mitigate the shrinkage imparted by the `Lasso` regularization parameter. This is done by performing an unrestricted least squares on the support selected by the `Lasso`. Of course the results will depend upon the tuning parameter used for the problem. We use two methods towards this end. In the first method we find the best `Lasso` solution (by obtaining an optimal tuning parameter based on minimizing

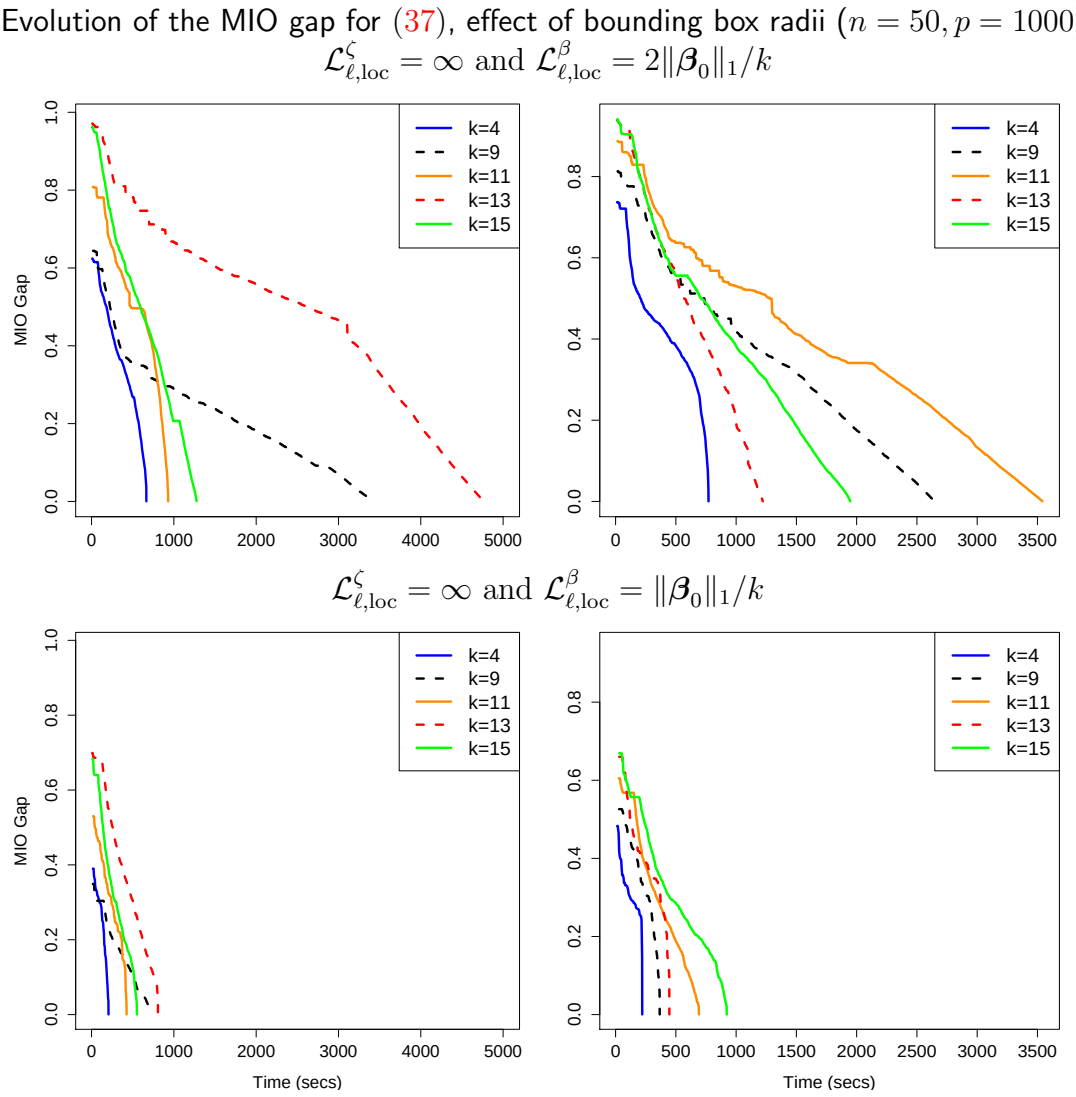

Figure 12: The evolution of the MIO gap with varying radii of bounding boxes for MIO formulation (37). The top panel has radii twice the size of the bottom panel. The dataset considered is generated as per Example 1 with $n = 50, p = 1000, \rho = 0.9$ and $k_0 = 5$ for different values of SNR: [Left Panel] SNR $= 1$, [Right Panel] SNR $= 3$. For each case, different values of k have been considered. The top panel has a bounding box radii which is twice the corresponding case in the lower panel. As expected, the times for the MIO gaps to close depends upon the radii of the boxes. The optimal solutions obtained were found to be insensitive to the choice of the bounding box radius.

predictive error on a held out validation set); we then obtain the un-regularized least squares solution for that `Lasso` solution. This typically performed worse than `Lasso` in all the experiments we tried—see Tables 3 and 4. The unrestricted least squares

solution on the optimal model selected by the `Lasso` (as shown in Figure 4) had worse predictive performance than the `Lasso`, with the same sparsity pattern, as shown in Table 3. This is probably due to overfitting since the model selected by the `Lasso` is quite dense compared to n, p. Table 4 presents the results for $50 = n \ll p = 1000$. We consider the same example presented in Figure 9, Example 1. First of all, Table 4 presents the prediction performance of `Lasso` after debiasing—we considered the same tuning parameter considered optimal for the `Lasso` problem. We see that as in the case of Table 3, the debiasing does not lead to improved performance in terms of prediction error.

We thus experimented with another variant of the debiased `Lasso`, where for every λ we computed the `Lasso` solution (2) and obtained $\hat{\boldsymbol{\beta}}_{\text{Deb},\lambda}$ by performing an unrestricted least squares fit on the support selected by the `Lasso` solution at λ. This method can be thought of delivering feasible solutions for Problem (1), for a value of $k := k(\lambda)$ determined by the `Lasso` solution at λ. The success of this method makes a case in support of using criterion (1). The tuning parameter was then selected by minimizing predictive performance on a held out test validation set. This method in general performed better than `Lasso` in delivering a sparser model with better predictive accuracy than the `Lasso`. The performance of the debiased `Lasso` was similar to `Sparsenet` and was in general inferior to MIO by orders of magnitude, especially for the problems where the pairwise correlations between the variables was large and SNR was low and $n \ll p$. The results are presented in Table 5,6 (for the case $n > p$) and 7 and 8 (for the case $n \ll p$).

Debiasing at optimal `Lasso` model, $n > p$

SNR	ρ	Ratio: `Lasso`/ Debiased `Lasso`
6.33	0.5	0.33
3.17	0.5	0.54
1.58	0.5	0.53
6.97	0.8	0.67
3.48	0.8	0.64
1.74	0.8	0.63
8.73	0.9	1
4.37	0.9	0.58
2.18	0.9	0.61

Table 3: `Lasso` and Debiased `Lasso` corresponding to the numerical experiments of Figure 4, for Example 1 with $n = 500, p = 100, \rho \in \{0.5, 0.8, 0.9\}$ and $k_0 = 10$. Here, "Ratio" equals the ratio of the prediction error of the `Lasso` and the debiased `Lasso` at the optimal tuning parameter selected by the `Lasso`.

Debiasing at optimal `Lasso` model, $n \ll p$

SNR	ρ	Ratio: `Lasso`/Debiased `Lasso`
10	0.8	0.90
7	0.8	1.0
3	0.8	0.91

Table 4: `Lasso` and Debiased `Lasso` corresponding to the numerical experiments of Figure 9, for Example 1 with $n = 50, p = 1000, \rho = 0.8$ and $k_0 = 5$. Here, "Ratio" equals the ratio of the prediction error of the `Lasso` and the debiased `Lasso` at the optimal tuning parameter selected by the `Lasso`.

The performance of this model was comparable with `Sparsenet`—it was better than `Lasso` in terms of obtaining a sparser model with better predictive accuracy. However, the performance of MIO was significantly better than the debiased version of the `Lasso`, especially for larger values of ρ and smaller SNR values.

Sparsity of Selected Models, $n > p$

SNR	ρ	`Lasso`	Debiased `Lasso`	MIO
6.33	0.5	27.6 (2.122)	10.9 (0.65)	10.8 (0.51)
3.17	0.5	27.7 (2.045)	10.9 (0.65)	10.1 (0.1)
1.58	0.5	28.0 (2.276)	10.9 (0.65)	10.2 (0.2)
6.97	0.8	34.1 (3.60)	10.4 (0.15)	10 (0.0)
3.48	0.8	34.0 (3.54)	10.9 (0.55)	10.2 (0.2)
1.74	0.8	33.7 (3.49)	13.7 (1.50)	10 (0.0)
8.73	0.9	25.9 (0.94)	13.9 (0.68)	10.5 (0.17)
4.37	0.9	34.6 (3.23)	18.1 (1.30)	10.2 (0.25)
2.18	0.9	34.7 (3.28)	20.5 (1.85)	10.1 (0.10)

Table 5: Number of non-zeros in the selected model by `Lasso`, Debiased `Lasso`, and MIO corresponding to the numerical experiments of Figure 4, for Example 1 with $n = 500, p = 100, \rho \in \{0.5, 0.8, 0.9\}$ and $k_0 = 10$. The tuning parameters for all three models were selected separately based on the best predictive model on a held out validation set. Numbers within brackets denote standard-errors. Debiased `Lasso` leads to less dense models than `Lasso`. When ρ is small and SNR is large, the model size of debiased `Lasso` performance is similar to MIO. However, for larger values of ρ and smaller values of SNR subset selection leads to orders of magnitude sparser solutions than debiased `Lasso`.

We then follow the method described above (for the $n > p$ case), where we consider a sequence of models $\hat{\boldsymbol{\beta}}_{\text{Deb},\lambda}$ and find the λ that delivers the best predictive model on a held out validation set.

Predictive Performance of Selected Models, $n > p$

SNR	ρ	`Lasso`	Debiased `Lasso`	MIO	Ratio: Debiased `Lasso`/MIO
6.33	0.5	0.0384 (0.001)	0.0255 (0.002)	0.0266 (0.001)	1.0
3.17	0.5	0.0768 (0.003)	0.0511 (0.004)	0.0478 (0.002)	1.0
1.58	0.5	0.1540 (0.007)	0.1021 (0.009)	0.0901 (0.009)	1.1
6.97	0.8	0.0389 (0.002)	0.0223 (0.001)	0.0231 (0.002)	1.0
3.48	0.8	0.0778 (0.004)	0.0464 (0.003)	0.0484 (0.004)	1.0
1.74	0.8	0.1557 (0.007)	0.1156 (0.008)	0.0795 (0.008)	1.5
8.73	0.9	0.0325 (0.001)	0.0220 (0.002)	0.0197 (0.002)	1.2
4.37	0.9	0.0632 (0.002)	0.0532 (0.003)	0.0427 (0.008)	1.3
2.18	0.9	0.1265 (0.005)	0.1254 (0.006)	0.0703 (0.011)	1.8

Table 6: Predictive Performance for tests of `Lasso`, Debiased `Lasso`, and MIO corresponding to the numerical experiments of Figure 4, for Example 1 with $n = 500, p = 100, \rho \in \{0.5, 0.8, 0.9\}$ and $k_0 = 10$. Numbers within brackets denote standard-errors. The tuning parameters for all three models were selected separately based on the best predictive model on a held out validation set. When ρ is small and SNR is large, debiased `Lasso` performance is similar to MIO. However, for larger values of ρ and smaller values of SNR subset selection performs better than debiased `Lasso` based solutions.

Sparsity of Selected Models, $n \ll p$

SNR	ρ	`Lasso`	Debiased `Lasso`	MIO
10	0.8	25.7 (1.73)	7.9 (0.43)	5 (0.12)
7	0.8	27.8 (2.69)	8.1 (0.43)	5 (0.16)
3	0.8	28.0 (2.72)	10.0 (0.88)	6 (1.18)

Table 7: Number of non-zeros in the selected model by `Lasso`, Debiased `Lasso`, and MIO corresponding to the numerical experiments of Figure 9, for Example 1with $n = 50, p = 1000, \rho = 0.8$ and $k_0 = 5$. Numbers within brackets denote standard-errors. The tuning parameters for all three models were selected separately based on the best predictive model on a held out validation set. Debiased `Lasso` leads to less dense models than `Lasso` but more dense models than MIO. The performance gap between MIO and debiased `Lasso` becomes larger with lower values of SNR.

Predictive Performance of Selected Models, $n \ll p$

SNR	ρ	`Lasso`	Debiased `Lasso`	MIO	Ratio: Debiased `Lasso`/ MIO
10	0.8	0.084 (0.004)	0.046 (0.003)	0.014 (0.005)	3.3
7	0.8	0.122 (0.005)	0.070 (0.004)	0.020 (0.007)	3.5
3	0.8	0.257 (0.012)	0.185 (0.016)	0.151 (0.027)	1.2

Table 8: Predictive performances of `Lasso`, Debiased `Lasso`, and MIO corresponding to the numerical experiments of Figure 9, for Example 1with $n = 50, p = 1000, \rho = 0.8$ and $k_0 = 5$. Numbers within brackets denote standard-errors. The tuning parameters for all three models were selected separately based on the best predictive model on a held out validation set. MIO consistently leads to better predictive models than Debiased `Lasso` and ordinary `Lasso`. Debiased `Lasso` performs better than ordinary `Lasso`.