SORT 45 (1) January-June 2021, 47-66 DOI: 10.2436/20.8080.02.109

Joint outlier detection and variable selection using discrete optimization

Mahdi Jammal[1], Stephane Canu[2] and Maher Abdallah[3]

Abstract

In regression, the quality of estimators is known to be very sensitive to the presence of spurious variables and outliers. Unfortunately, this is a frequent situation when dealing with real data. To handle outlier proneness and achieve variable selection, we propose a robust method performing the outright rejection of discordant observations together with the selection of relevant variables. A natural way to define the corresponding optimization problem is to use the ℓ_0 norm and recast it as a mixed integer optimization problem. To retrieve this global solution more efficiently, we suggest the use of additional constraints as well as a clever initialization. To this end, an efficient and scalable non-convex proximal alternate algorithm is introduced. An empirical comparison between the ℓ_0 norm approach and its ℓ_1 relaxation is presented as well. Results on both synthetic and real data sets provided that the mixed integer programming approach and its discrete first order warm start provide high quality solutions.

MSC: 62J05, 62J20, 62J07, 62G35, 90C11, 68T05.

Keywords: Robust optimization, statistical learning, linear regression, variable selection, outlier detection, mixed integer programming.

1 Introduction

We consider the linear regression model:

$$y = X\beta + \epsilon.$$

where $y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n\times p}$ is the model matrix, $\beta \in \mathbb{R}^p$ is the vector of regression coefficients and $\epsilon \in \mathbb{R}^n$ is the error vector. It is convenient to estimate β with a sparse vector, especially for high values of p.

[1] Institut National des Sciences Appliquees (INSA) de Rouen 685 Avenue de l'Universite 76800 Saint-Etienne du Rouvray; Lebanese Univerity, Beirut, Lebanon.

[2] Institut National des Sciences Appliquees (INSA) de Rouen 685 Avenue de l'Universite 76800 Saint-Etienne du Rouvray.

[3] Lebanese University, Faculty of public health, Hadath, Beirut, Lebanon.

Received: October 2020
Accepted: April 2021

It is well known that dimension reduction or feature selection is an effective strategy to handle contaminated data and to deal with high dimensionality while providing better prediction (Bertsimas, King and Mazumder, 2015). Outliers, i.e. atypical or corrupted observations, can also have a considerable bad influence on estimators (Yang et al., 2010; Rousseeuw and Hubert, 2018). Usually, outliers are eliminated in a time consuming data cleaning pretreatment (Hodge and Austin, 2004; Campos et al., 2016) while variable selection is performed together with parameter estimation using the Lasso (Tibshirani, 1996), its variants (Tibshirani, Wainwright and Hastie, 2015) or the best subset (Bertsimas et al., 2015) algorithms just to name a few. For a recent comparison of these algorithms, see for instance Hastie, Tibshirani and Tibshirani (2017). However, it is well known that, due to the ordinary least square (OLS) criterion used in the lasso, it is not robust to outliers. For instance, Alfons et al. (2013) show that the breakdown point of the lasso is $1/n$, that is, only one single outlier can make the lasso estimate completely unreliable.

Different attempts have been made to solve this problem by mixing variable selection and outlier detection. A popular idea is to replace the OLS criterion of the lasso by a loss robust to outliers such as the absolute deviation (Wang, Li and Jiang, 2007), the least trimmed squares estimator (Alfons et al., 2013) introduced by Rousseeuw and Leroy (1987) or the Huber's loss (Dalalyan and Thompson, 2019). Also, to deal with the specific case of cellwise contamination, that is the presence of outliers in the design matrix, Öllerer, Alfons and Croux (2016) introduced the shooting S-estimator.

However, none of these approaches considered the use of the pseudo ℓ_0 norm as recently introduced by Bertsimas et al. (2015). In this paper we propose to get robust estimates by solving these two problems of variable selection and outliers detection together using pseudo ℓ_0 norms for both. Such an approach leads to reformulating the double robust regression problem as a mixed integer program providing a global solution with convergence guarantee in case of early stopping as well as flexibility and adaptability. It also allows the use of efficient solvers such as Gurobi, the one used in our experiments to obtain good results on both synthetic and real data.

Brief Context and Background

Let $X = (x_1, \ldots, x_n)^\intercal$ be a $n \times p$ design matrix and $y \in \mathbb{R}^n$ a response vector. We consider the following linear model to accommodate outliers:

$$\forall i \in \{1, \ldots, n\}, \qquad y_i = \begin{cases} x_i^\intercal \beta + \epsilon_i & \text{if observation } i \text{ is regular} \\ \gamma_i & \text{if observation } i \text{ is an outlier to be trimmed,} \end{cases} \tag{1}$$

where $\beta \in \mathbb{R}^p$ is the unknown parameter vector to be estimated, $\epsilon \in \mathbb{R}^n$ is the noise vector and $\gamma \in \mathbb{R}^n$ an intervention vector. A way to model doubtful observations to be trimmed is to introduce a vector $\tau \in \mathbb{R}^n$ modeling outliers:

$$\forall i \in \{1,\dots,n\}, \qquad \tau_i = \begin{cases} 0 & \text{if observation } i \text{ has to be taken into account} \\ y_i - x_i^\top \beta - \epsilon_i & \text{if observation } i \text{ is an outlier to be trimmed,} \end{cases}$$

The model (1) can be rewritten as the following linear model She and Owen (2011):

$$y = X\beta + \epsilon + \tau. \tag{2}$$

We are interested in minimizing the norm of the noise vector while selecting k_v variables and removing k_o outliers, that is, solving the following optimization problem Chen, Caramanis and Mannor (2013), for some $q \in \{1,2\}$,

$$\begin{array}{ll} \min\limits_{\beta\in\mathbb{R}^p,\tau\in\mathbb{R}^n} & \frac{1}{q}\|X\beta+\tau-y\|_q^q \\ \text{s.t.} & \|\beta\|_0 \le k_v \\ & \|\tau\|_0 \le k_o, \end{array} \tag{3}$$

This formulation allows the selection of relevant variables and the avoidance of outliers. When $k_o = 0$, no outlier detection is performed and this problem boils down to the best subset selection problem Miller (2002); Bertsimas et al. (2015); Miyashiro and Takano (2015). When $k_v = p$, no variable selection is performed, the resulting problem is known as the least trimmed squares regression problem Rousseeuw and Leroy (1987); Giloni and Padberg (2002). Due to the nature of the cardinality constraints, Problem (3) is a non-convex optimization problem and has been shown to be NP-hard and considered as an intractable problem. Mainstream research focused on solving a relaxed version of Problem (3), by using the ℓ_1 norm instead of the ℓ_0 norm:

$$\begin{array}{ll} \min\limits_{\beta\in\mathbb{R}^p,\tau\in\mathbb{R}^n} & \frac{1}{2}\|X\beta+\tau-y\|_2^2 \\ \text{s.t.} & \|\beta\|_1 \le \lambda \\ & \|\tau\|_1 \le \gamma \end{array} \tag{4}$$

where λ and γ are two nonnegative regularization parameters. Problem (4) will be denoted by ℓ_1-RR. However, this approach is not globally optimal in the sense of (3) since it will not necessarily provide the same solution provided by (3). We recall that the lagrangian relaxation of Problem (4) is given by:

$$\min_{\beta\in\mathbb{R}^p,\tau\in\mathbb{R}^n} \quad \frac{1}{2}\|X\beta+\tau-y\|_2^2 + \lambda||\beta||_1 + \gamma||\tau||_1 \tag{5}$$

Statistical properties of Problem (5) have been explored in Dalalyan and Thompson (2019); Nguyen and Tran (2013). To retrieve the global minimum of Problem (3), we propose to recast Problem (3) as a mixed integer optimization problem (MIO), which allows the use of efficient solvers to solve it, "Gurobi" for example. The MIO approach has a computational cost, but two decades of progress enabled its effective practical use for moderately sized problems. We also present a discrete first order algorithm that pro-

vides a high quality solution that could be used as a warm start for the MIO algorithm. In addition, it is useful for high-dimensional data sets since it provides solutions in a short time.

The remainder of the paper is organized as follows. In Section 2, we present our approach for variable selection and outliers detection using the ℓ_0 together with its formulation as a mixed integer optimization allowing to obtain the global solution. Section 3 introduces a relaxation that provides efficiently a local solution to this problem. This is followed by Sections 4 and 5 reporting empirical evidence on both synthetic and real data sets respectively. Finally, the paper is concluded in Section 6.

2 Variable Selection and Outlier Detection as a MIO

We propose to reformulate Problem (3) as a mixed integer (binary) optimization (MIO) problem by introducing binary variables representing whether or not variables and observations are useful.

2.1 Introducing Binary Variables

Variable selection involves the ℓ_0 norm function to count the number of useful variables. This counting function can be represented by introducing p binary variables $z_j \in \{0,1\}$ such that

$$\|\beta\|_0 = \sum_{j=1}^{p} z_j \quad \text{and} \quad z_j = 0 \Rightarrow \beta_j = 0.$$

Different approaches can be used to force $z_j = 0 \Leftrightarrow \beta_j = 0$ into an optimization problem, such as:

1. Replace β_j by $z_j\beta_j$ for $j = 1,\ldots,p$.
2. Set $|\beta_j|(1-z_j) = 0$ for $j = 1,\ldots,p$ or $\sum_{j=1}^{p} |\beta_j|(1-z_j) = 0$.
3. Use a big-M constraint, $|\beta_j| \leq M_v z_j$ for $j = 1,\ldots,p$ and for some fixed constant M_v large enough (such as $M_v \geq \max_j |\beta_j^\star|$, $\beta_j^\star$ being the solution of the optimization problem). In the setup of experimental resuls for synthetic data sets, we explain how we can set a priori value of M_v.
4. Treat $z_j = 0 \Leftrightarrow \beta_j = 0$ as logical implications (also called indicator constraints or special ordered set SOS-1). Note that this kind of logical implication can be efficiently handled in a branch-and-bound procedure for MIO problems.

We now discuss and give a short overview of the advantages and drawbacks of each approach. The two first approaches involve nonlinear interaction terms between binary and continuous variables. Their interest lies in the possibility of obtaining interesting

continuous relaxations. The main advantage of the big-M method (approach 3) is that it brings only linear inequality constraints, but the value of the M term needs to be chosen carefully since it shows a great deal of practical influence on the solver performance. Logical implications (approach 4) have the advantage of avoiding these types of problems, as they do not rely on a separate constant value. However, they tend to have weaker relaxations, a condition which may lead to longer solve times in a model. In this paper we will use the third approach for our implementation since the presented discrete first order algorithm allows to obtain a good upper bound of M and since the brought linear inequality constraint do not have a significant influence on the computational time.

Outlier detection also involves the ℓ_0 norm function to count the number of outliers. As done above, this counting function can be represented by introducing n binary variables $t_i \in \{0,1\}$ such as

$$\|\tau\|_0 = \sum_{i=1}^{n} t_i \quad \text{and} \quad t_i = 0 \Rightarrow \tau_i = 0, \ (x_i, y_i) \text{ is not an outlier.}$$

2.2 A MIO Formulation

Introducing binary variables for both variables and outliers with two big-M constraints, given appropriate parameters k_v, k_o, M_v and M_o, Problem (3) becomes for some $q \in \{1,2\}$:

$$\begin{aligned} \min_{\beta\in\mathbb{R}^p, \tau\in\mathbb{R}^n, z\in\{0,1\}^p, t\in\{0,1\}^n} \quad & \tfrac{1}{q}\|X\beta + \tau - y\|_q^q & & \\ \text{s.t.} \quad & \sum_{j=1}^{p} z_j \le k_v \quad \text{and} \quad |\beta_j| \le z_j M_v, & & j = 1,\dots,p \\ & \sum_{i=1}^{n} t_i \le k_o \quad \text{and} \quad |\tau_i| \le t_i M_o & & i = 1,\dots,n. \end{aligned} \tag{6}$$

This problem turns out to be a mixed binary quadratic program when $q = 2$, it will be denoted by ℓ_0-RR and it will be used in the rest of the paper. However, we will introduce other formulations that could also be efficient without using these formulations in the experiments.

2.3 Convergence to the Global Optimum

Figure (1) shows the influence of the SNR value on the speed of convergence. In fact, we consider a synthetic data set without adding outliers. When $k_o = 5\%$, the time needed to certify the optimality decreased from 120 seconds for $SNR = 0.5$ to 52 seconds for $SNR = 5$. In addition, after three hours of computation and when $k_o = 10\%$, the MIO-Gap decreased from 0.2 ($SNR = 0.5$) to 0.1 ($SNR = 5$).

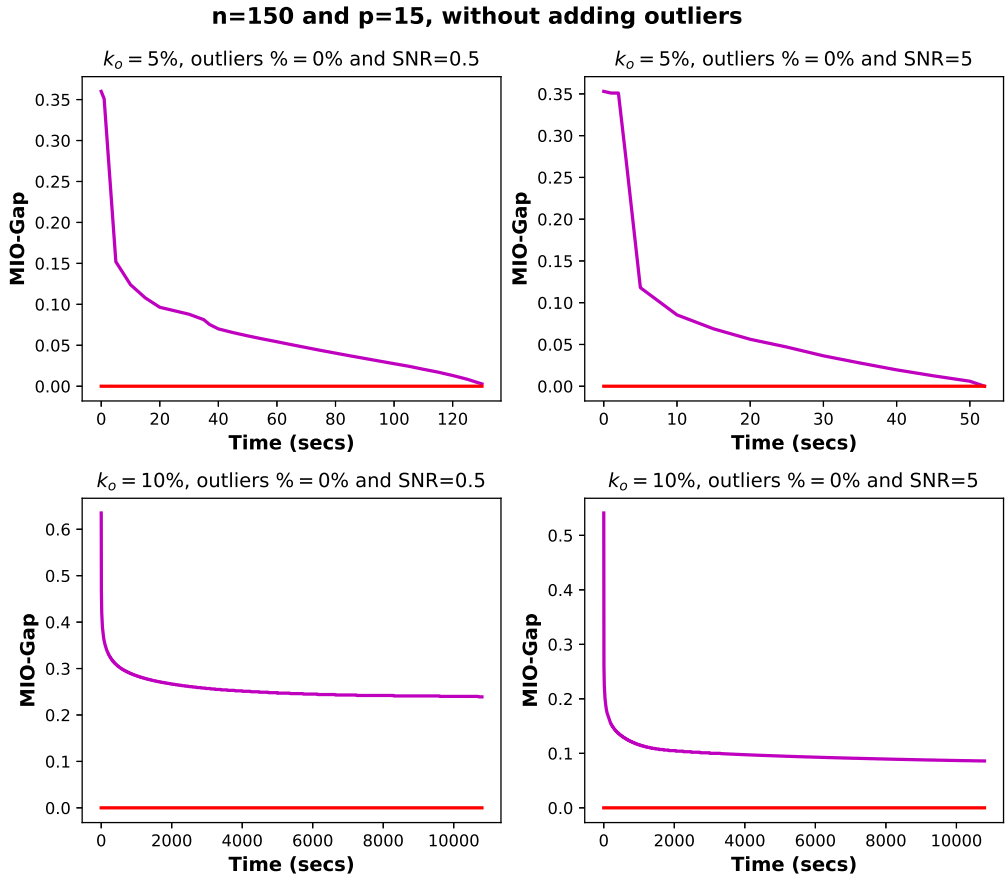

Figure 1: *The typical evolution of the MIO formulation (6) for a synthetic dataset with* $n = 150, p = 15, s = 5$. *The top and the bottom panels show the evolution of the corresponding MIO gap with time. The red line is the* $y = 0$ *reference line.*

In Figure (2) we shed light on the importance of estimating the true percentage of outliers in the data set (10% in our case). When we set k_o as the true percentage of outliers (right panel), the optimality was certified in about three minutes. But when the true percentage of outliers is underestimated ($k_o = 2.5\%$), the MIO-Gap was still about 0.2 even after 3 hours. Note that when we overestimate the percentage of outliers ($k_o = 15\%$ for example) we observe slow convergence as we did when underestimating it.

In summary, the convergence rate depends on many factors:

- the size of the data set: smaller data leads to faster convergence to optimality,
- the estimation of the parameters k_v and k_o: better estimation of the number of relevant features and of the percentage of outliers increases the speed of convergence to optimality,
- the noise in the data (SNR): more time is needed to certify optimality for lower SNR values.

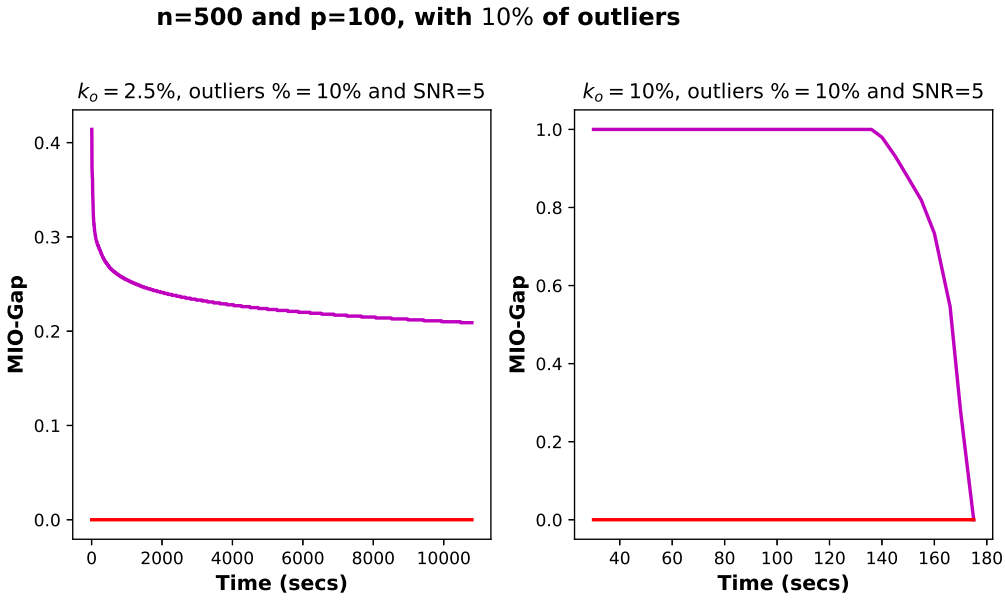

Figure 2: *The typical evolution of the MIO formulation (6) for a synthetic dataset with* $n = 500, p = 100, s = 5$. *The left and the right panels show the evolution of the corresponding MIO gap with time. The red line is the* $y = 0$ *reference line.*

3 Proximal Alternating Linearized Minimization Algorithm

In this section, an efficient alternate projected gradient algorithm providing a local solution to the optimization Problem (3) is introduced. This algorithm will be used as a warm-start procedure for the MIO solver as well as an optimization algorithm itself since it could provide high quality solutions in a short time. Before entering into the details of the alternate projected gradient algorithm, it is appropriate to introduce the problem of finding the projection of a vector $\mathrm{u} \in \mathbb{R}^p$ onto the set of $k \leq p$ sparse vectors

$$\begin{array}{ll} \min\limits_{\mathrm{v}\in\mathbb{R}^p} & \frac{1}{2}\|\mathrm{v}-\mathrm{u}\|^2 \\ \text{s.t.} & \|\mathrm{v}\|_0 \leq k. \end{array} \tag{7}$$

This problem is easy and its solution $\mathrm{v}^\star$ is given by sorting on the absolute value of vector $|\mathrm{u}|$, that is by a sequence of indices (j) such that $|u_{(1)}| \geq |u_{(2)}| \geq \ldots |u_{(j)}| \geq \cdots \geq |u_{(p)}|$. Using these indices, the projection $\mathrm{v}^\star = P_k(\mathrm{u})$ of u is the vector u itself with its smallest coefficients set to 0 that is

$$\mathrm{v}^\star = P_k(\mathrm{u}) = \left\{ \begin{array}{ll} u_j & \text{if } j \in \{(1),\ldots,(k)\} \\ 0 & \text{else.} \end{array} \right. \tag{8}$$

We propose to use this projection mechanism, on both β and τ, to get a solution to the initial Problem (3) at a low computing cost.

A possible way to achieve this goal consists of using the so-called block Gauss-Seidel iteration scheme on variables β and τ, also known as alternating minimization. To this end, a sequence $\left\{(\beta^\ell,\tau^\ell)\right\}_{\ell\in\mathbb{N}}$ is generated starting from some (β^0,τ^0) using the following scheme:

$$\left\{\begin{array}{ll}\beta^{\ell+1}=\underset{\beta\in\mathbb{R}^p}{\arg\min} & (\beta-\beta^\ell)^\mathsf{T}X^\mathsf{T}(X\beta^\ell+\tau^\ell-y)\\ \text{s.t.} & \|\beta\|_0\leq k_v\\ & \|\beta-\beta^\ell\|^2\leq d_v\end{array}\right.\quad\left\{\begin{array}{ll}\tau^{\ell+1}=\underset{\tau\in\mathbb{R}^n}{\arg\min} & (\tau-\tau^\ell)^\mathsf{T}(X\beta^{\ell+1}+\tau^\ell-y)\\ \text{s.t.} & \|\tau\|_0\leq k_o\\ & \|\tau-\tau^\ell\|^2\leq d_o.\end{array}\right. \tag{9}$$

$$\left\{\begin{array}{l}\frac{1}{2}\|X\beta+\tau^\ell-y\|^2\leq\frac{1}{2}\|X\beta^\ell+\tau^\ell-y\|^2+(\beta-\beta^\ell)^\mathsf{T}X^\mathsf{T}(X\beta^\ell+\tau^\ell-y)+\frac{1}{2\rho_v}\|\beta-\beta^\ell\|^2\\ \frac{1}{2}\|X\beta^{\ell+1}+\tau-y\|^2\leq\frac{1}{2}\|X\beta^{\ell+1}+\tau^\ell-y\|^2+(\tau-\tau^\ell)^\mathsf{T}(X\beta^{\ell+1}+\tau^\ell-y)+\frac{1}{2\rho_o}\|\tau-\tau^\ell\|^2.\end{array}\right. \tag{10}$$

Where d_v and d_o are two given positive parameters that can be changed each step. The idea of the proximal method is, at each iteration, to minimize a regularized first-order approximation of the cost that can be interpreted as a local trust region mechanism (for details see for instance Parikh and Boyd, 2014). This surrogate loss is also a local upper bound of the targeted loss since, for well chosen ρ_v and ρ_o, the Lagrange multipliers associated with the trust region constraints

$$\left\{\begin{array}{l}\frac{1}{2}\|X\beta+\tau^\ell-y\|^2\leq\frac{1}{2}\|X\beta^\ell+\tau^\ell-y\|^2+(\beta-\beta^\ell)^\mathsf{T}X^\mathsf{T}(X\beta^\ell+\tau^\ell-y)+\frac{1}{2\rho_v}\|\beta-\beta^\ell\|^2\\ \frac{1}{2}\|X\beta^{\ell+1}+\tau-y\|^2\leq\frac{1}{2}\|X\beta^{\ell+1}+\tau^\ell-y\|^2+(\tau-\tau^\ell)^\mathsf{T}(X\beta^{\ell+1}+\tau^\ell-y)+\frac{1}{2\rho_o}\|\tau-\tau^\ell\|^2.\end{array}\right. \tag{11}$$

For each iteration, this method introduced by Bolte, Sabach and Teboulle (2014) and called the proximal alternating linearized minimization (PALM) algorithm, consists of minimizing the upper bounds as follows:

$$\left\{\begin{array}{l}\beta^{\ell+1}=\underset{\beta\in\mathbb{R}^p,\|\beta\|_0\leq k_v}{\arg\min}\ (\beta-\beta^\ell)^\mathsf{T}X^\mathsf{T}(X\beta^\ell+\tau^\ell-y)+\frac{1}{2\rho_v}\|\beta-\beta^\ell\|^2\\ \tau^{\ell+1}=\underset{\tau\in\mathbb{R}^n,\|\tau\|_0\leq k_o}{\arg\min}\ (\tau-\tau^\ell)^\mathsf{T}(X\beta^{\ell+1}+\tau^\ell-y)+\frac{1}{2\rho_o}\|\tau-\tau^\ell\|^2.\end{array}\right. \tag{12}$$

That is, after some algebra,

$$\left\{\begin{array}{ll}\beta^{\ell+1}=\underset{\beta\in\mathbb{R}^p,\|\beta\|_0\leq k_v}{\arg\min} & \frac{1}{2}\|\beta-\beta^\ell+\rho_vX^\mathsf{T}(X\beta^\ell+\tau^\ell-y)\|^2\\ \tau^{\ell+1}=\underset{\tau\in\mathbb{R}^n,\|\tau\|_0\leq k_o}{\arg\min} & \frac{1}{2}\|\tau-\tau^\ell+\rho_o(X\beta^{\ell+1}+\tau^\ell-y)\|^2.\end{array}\right. \tag{13}$$

These two minimization problems are of the same kind as Problem (7) and thus the sequence can be generated by using two ℓ_0 projected gradient, that is:

$$\left\{\begin{array}{l}\beta^{\ell+1}=P_{k_v}\left(\beta^\ell-\rho_vX^\mathsf{T}(X\beta^\ell+\tau^\ell-y)\right)\\ \tau^{\ell+1}=P_{k_o}\left(\tau^\ell-\rho_o(X\beta^{\ell+1}+\tau^\ell-y)\right).\end{array}\right. \tag{14}$$

Algorithm 1 presents the pseudo code of the PALM algorithm.

Algorithm 1: Proximal alternating linearized minimization (PALM) Bolte et al. (2014)

Data: X, y initialization $\beta, \tau = 0$
Result: β, τ
set $\rho_v \leq \frac{1}{\sigma_M^2}$ and $\rho_o \leq 1$
while *it has not converged* $(||\beta_{n+1} - \beta_n||_2 > 10^{-6})$ **do**
  $\mathrm{d} \leftarrow \beta - \rho_v X^\top (X\beta + \tau - y)$ — variable selection
  $\beta \leftarrow P_{k_v}(\mathrm{d})$
  $\delta \leftarrow \tau - \rho_o (X\beta + \tau - y)$ — eliminating outliers
  $\tau \leftarrow P_{k_o}(\delta)$

This algorithm converges towards a local minima of Problem (3) since it fulfills the assumptions needed for Theorem 3.1 in Bolte et al. (2014). Indeed, if we consider $G(\beta, \tau) = \frac{1}{2}||X\beta + \tau - y||_2^2$, PALM converges if the partial gradients $G_\beta(\beta) = X^\top (X\beta + \tau - y)$ and $G_\tau(\tau) = (X\beta + \tau - y)$ are globally Lipschitz with modules L_1 and L_2 respectively. It could be easily shown that $G_\beta(\beta)$ and $G_\tau(\tau)$ are $\frac{1}{\sigma_M^2}$ and 1 Lipschitz respectively, σ_M being the largest singular value of X. Thus the step sizes could be chosen such that $\rho_v \leq \frac{1}{\sigma_M^2}$ and $\rho_o \leq 1$ as proved in Bolte et al. (2014).

4 Results for Synthetic Data Sets

In this section we show the empirical performance of the MIO approach.

4.1 Setup

In Hastie et al. (2017), a follow-up paper to Bertsimas et al. (2015), the authors provide a synthetic setup considering a wide range of SNR values. We use it here to compare the best subset selection (Formulation (6) with $k_o = 0$), the lasso, PALM, the ℓ_0 robust regression - ℓ_0 RR and the ℓ_1 robust regression - ℓ_1 RR. The same notations as Hastie et al. (2017) were used, namely n, p (problem dimensions), s (sparsity level), beta-type (pattern of sparsity), ρ (predictor auto-correlation level) which controls correlations between predictor variables, and ν (SNR level).

- We define coefficients $\beta_0 \in \mathbb{R}^p$ according to s and the beta-type, as described below.
- We draw the rows of the matrix $X \in \mathbb{R}^{n \times p}$ from $N_p(0, \Sigma)$, where $\Sigma \in \mathbb{R}^{p \times p}$ has entry (i, j) equal to $\rho^{|i-j|}$, and $\rho = 0.35$.

- We draw the vector $y \in \mathbb{R}^n$ from $N_n(X\beta_0, \sigma^2 I)$, with σ^2 defined to meet the desired SNR level, i.e., $\sigma^2 = \beta_0^T \Sigma \beta_0 / \nu$.
- We use 5-fold cross-validation and the tuning was performed by minimizing prediction error on the test set.
- To assess the influence of outliers, 5% of outliers were added to the data set by following a normal $N(50, \sigma)$ instead of $N(0, \sigma)$.
- We considered two configurations: the low setting with $n = 150, p = 15$, and the medium setting $n = 500, p = 100$. For each configuration, we also considered two settings: the first one with outliers generated as mentioned above, and the second one without adding outliers.
- The lasso was tuned over 100 values of λ (as it is in `glmnet`).
- In order to determine the values of k_v, M_v, k_o and M_o, we run the PALM algorithm for k_v ranging from 1 to p and for k_o ranging from 0 to 10% with a step size of 2.5%. Then, we choose the solution with the minimal error $||X_{test}\beta_{palm} - y_{test}||_2^2$.
- $M_v = (1+\alpha)||\beta_{palm}||_\infty$, $M_o = (1+\alpha)||\tau_{palm}||_\infty$ with $\alpha = 0.1$, k_v and k_o are set as the number of nonzero elements in the solutions β_{palm} and τ_{palm} respectively.
- The ℓ_1 robust regression (ℓ_1 RR) algorithm was tuned over five values of λ from zero to $1.5||\beta_{lasso}||_\infty$ where β_{lasso} is the solution obtained by the lasso method, and over fifty one values of γ from 0 to 5000 with a step size of 100 for the low dimensional case, and from 0 to 10000 with a step size of 200 for the medium dimensional case.
- We run the best subset selection, the lasso, PALM, the ℓ_0 robust regression (ℓ_0 RR) the ℓ_1 robust regression (ℓ_1 RR) using a 5-fold cross-validation. The tuning was performed by minimizing the error on the test set.
- We repeat 10 times for the low dimensional setting and 5 times for the medium dimensional setting and average the results.

Coefficients: We considered three settings for the coefficients $\beta_0 \in \mathbb{R}^p$ as in Hastie et al. (2017):

- beta-type 1: $\beta_0 = (1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ 1,\ 0,\ \underbrace{0,\ldots,0}_{p-10 \text{ times}})$;
- beta-type 2: β_0 has its first 5 components equal to 1, and the rest equal to 0;
- beta-type 5: β_0 has its first 5 components equal to 1, and the rest decaying exponentially to 0, specifically, $\beta_{0i} = 0.5^{i-s}$, for $i = s+1 \ldots p$, where $s = 5$;

Following Bertsimas et al. (2015); Hastie et al. (2017), we use, as an accuracy metric, the relative risk (R.R) defined by:

$$\text{R.R}(\hat{\beta}) = \frac{\mathbb{E}(x_0^T\hat{\beta} - x_0^T\beta_0)^2}{\mathbb{E}(x_0^T\beta_0)^2} = \frac{(\hat{\beta}-\beta_0)^T\Sigma(\hat{\beta}-\beta_0)}{\beta_0^T\Sigma\beta_0},$$

The best score is 0 (when $\hat{\beta} = \beta_0$) and the null score is 1, obtained when $\hat{\beta} = 0$.
We also use the proportion of variance explained (PVE) defined by:

$$\text{PVE}(\hat{\beta}) = 1 - \frac{\mathbb{E}(y_0 - x_0^T\hat{\beta})^2}{\text{Var}(y_0)} = 1 - \frac{(\hat{\beta} - \beta_0)^T\Sigma(\hat{\beta} - \beta_0) + \sigma^2}{\beta_0^T\Sigma\beta_0 + \sigma^2}.$$

The maximum value for the PVE, also called the perfect score, is SNR/(SNR+1) (see Hastie et al. (2017) for details).

4.2 Computational Costs

For the lasso, we used the **Matlab** "lasso" function with 100 values of λ as implemented in **glmnet**. The solution is delivered in a very short time. For the best subset selection problem, we implemented the method using the MIO Formulation (6) with $k_o = 0$, used PALM to compute a warm start and then call Gurobi through its Matlab interface. We used a time limit of 3 minutes for Gurobi to optimize the best subset selection problem for both low and medium dimensional case. The same procedure is followed for the ℓ_0 robust regression problem but with a time limit increased to 10 minutes for the medium dimensional setting.

For the ℓ_1 robust regression, we obtained $5 \times 51 = 255$ (5 values of λ and 51 values of γ) solutions for each test. The time needed to obtain each solution depends on the size of the dataset, but it varies from 0.16 second to about 1 second.
We can conclude that for low dimensional setting, we faced around 15 hours of computation (10 repetitions), and more than 45 hours for the medium dimensional setting (5 repetitions) for each type of β. Using only one cross-validation loop would decrease significantly the computational time of the experiments. We note that the computations were carried on in a windows 10 64-bit server - Intel(R) Core(TM) i7-4700MQ CPU @ 2.40 GHz and 8 GB of Ram. So using a more powerful machine would help to decrease the computational cost.

4.3 Results

Figures (3)-(8) plot the relative risk (left panel) and the proportion of variance explained (right panel) as functions of signal-to-noise ratio (SNR). The results can be divided into two main categories:

4.3.1 No Outliers

In this case, no outliers were added to the synthetic data sets generated as mentioned before. Figures (3), (4), (5), (6), (7) and (8) show that for small SNR values, the ℓ_1 methods (lasso and ℓ_1 RR) have the lead on the other methods (best subset selection, PALM and ℓ_0 RR). While for high SNR values the ℓ_0 approaches outperform the ℓ_1

approaches even though all the methods perform quite similarly for high SNR values. These results shed the light on the capability of the MIO approach to perform well when no outliers exist in the data set.

4.3.2 Presence of Outliers

In this case, Figures (3), (4), (5), (6), (7) and (8) show that PALM, ℓ_0 RR and ℓ_1 RR outperform the best subset selection and the lasso, which is not surprising since the last two methods are not robust to outliers. The obtained results ensure that adding the variable τ helped to improve the performance of the estimators and guaranteed obtaining robust methods. In addition, for $SNR < 0.25$ the ℓ_1 RR performs, in general, better than PALM and the ℓ_0 RR. But for higher SNR values, there is no clear winner. An important caveat to emphasize up front is that the Gurobi MIO algorithm for ℓ_0 RR was given only

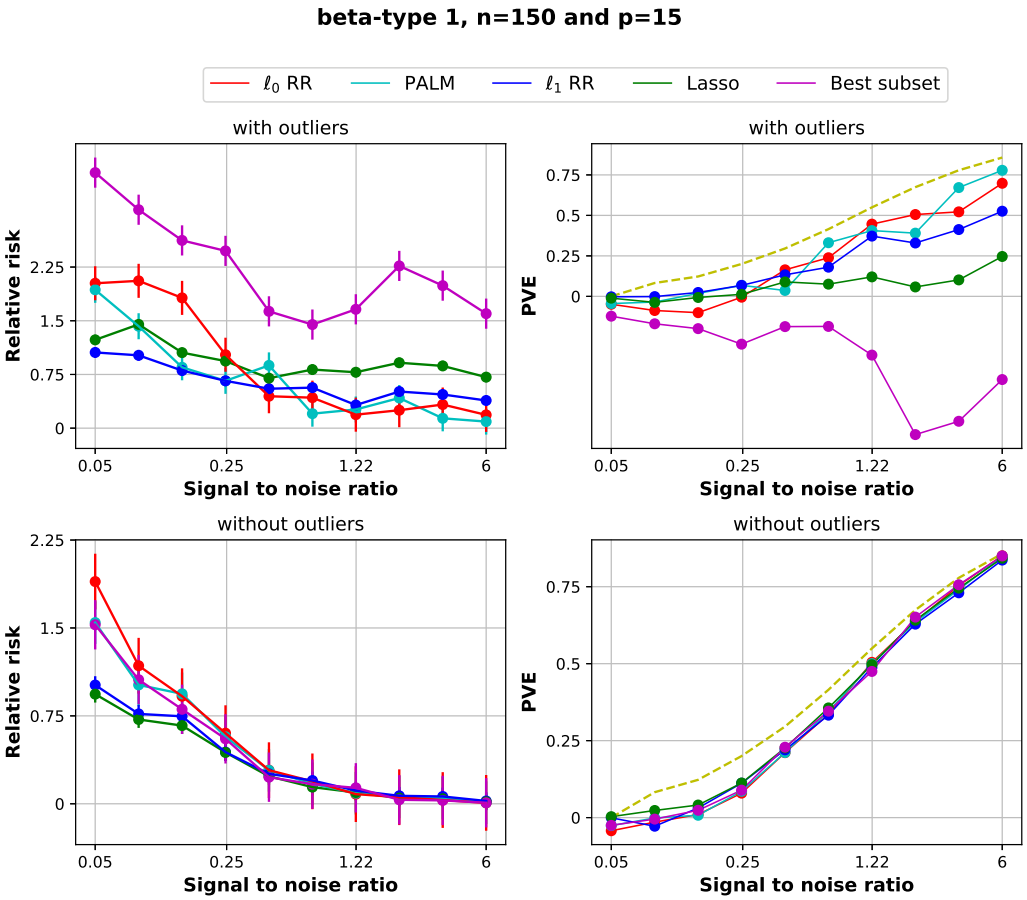

Figure 3: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 1 in the setting with $n = 150$, $p = 15$, and $s = 5$ with and without outliers (top panel and bottom panel respectively).*

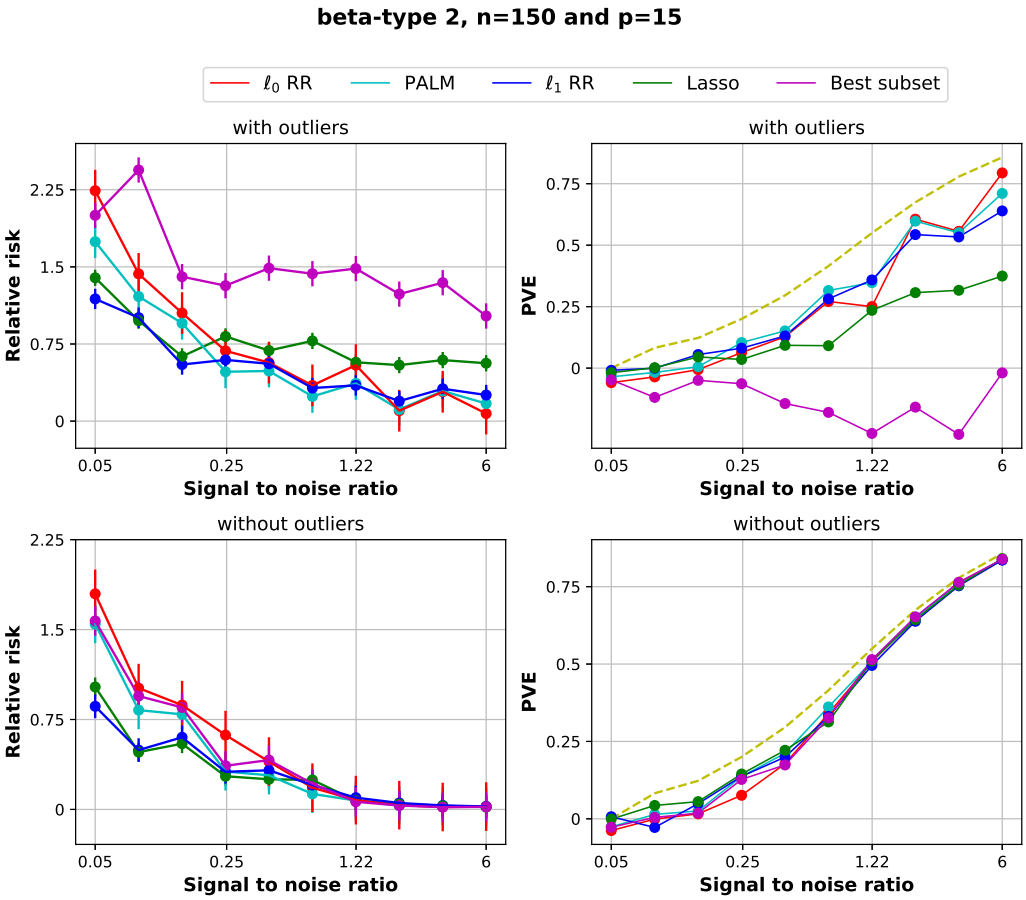

Figure 4: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 2 in the setting with $n = 150$, $p = 15$, and $s = 5$ with and without outliers (top panel and bottom panel respectively).*

10 minutes per problem, which may have caused the ℓ_0 RR to underperform, and that the performance of the MIO algorithm depends on the parameters tuned using PALM.

4.4 Detection Rate for the Feature Selection and Outlier Detection Tasks

To determine whether the ℓ_0 robust regression approach can detect the outliers and select the right features, we generated two low-dimensional and two medium-dimensional data sets using the β type-2, with SNR values 0.5 and 5. We added 5% of outliers in the response vector (as in the setup of the synthetic data sets). k_v and k_o were set as the true sparsity level of β and as the percentage of outliers (5%). In all cases, the detection rate of both outliers and features was 100%, noting that no cross-validation was performed.

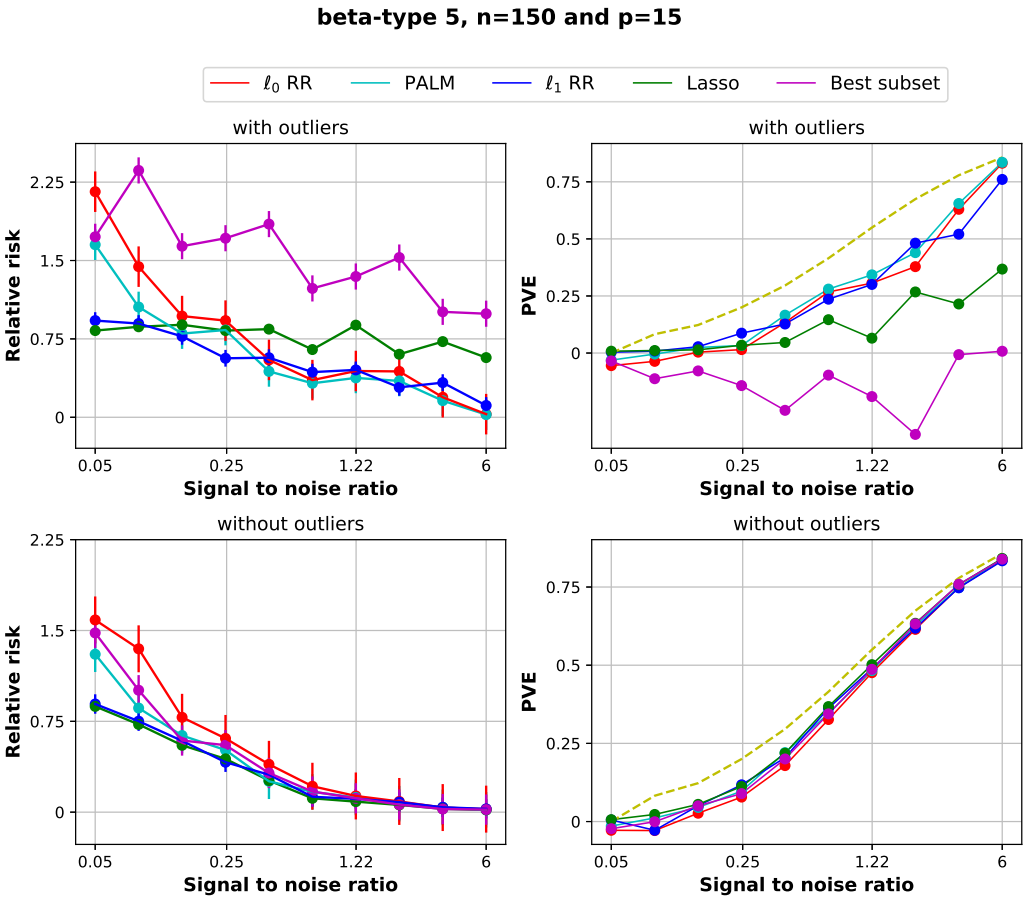

Figure 5: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 5 in the setting with* $n = 150$*,* $p = 15$*, and* $s = 5$ *with and without outliers (top panel and bottom panel respectively).*

In the experiments performed on both real and data sets, we used PALM to tune the parameters k_v and k_o. Thus the performance of the MIO approach depends on PALM. To this end, each data set was split into two parts: the training set (70%) and the testing set (30%). We added 5% of outliers in the training set's response vector. PALM was performed for $k_v \in [1,\ldots,p]$ and $\frac{k_o}{n} \in [0, 0.025, 0.05, 0.075, 0.1]$. PALM failed to estimate the true sparsity level and the true percentage of outliers as seen in Figures (9) and (10). This leads the PALM-MIO approach to fail at detecting the percentage of outliers and selecting the correct number of relevant features, even though all the true outliers were considered as outliers by this approach.

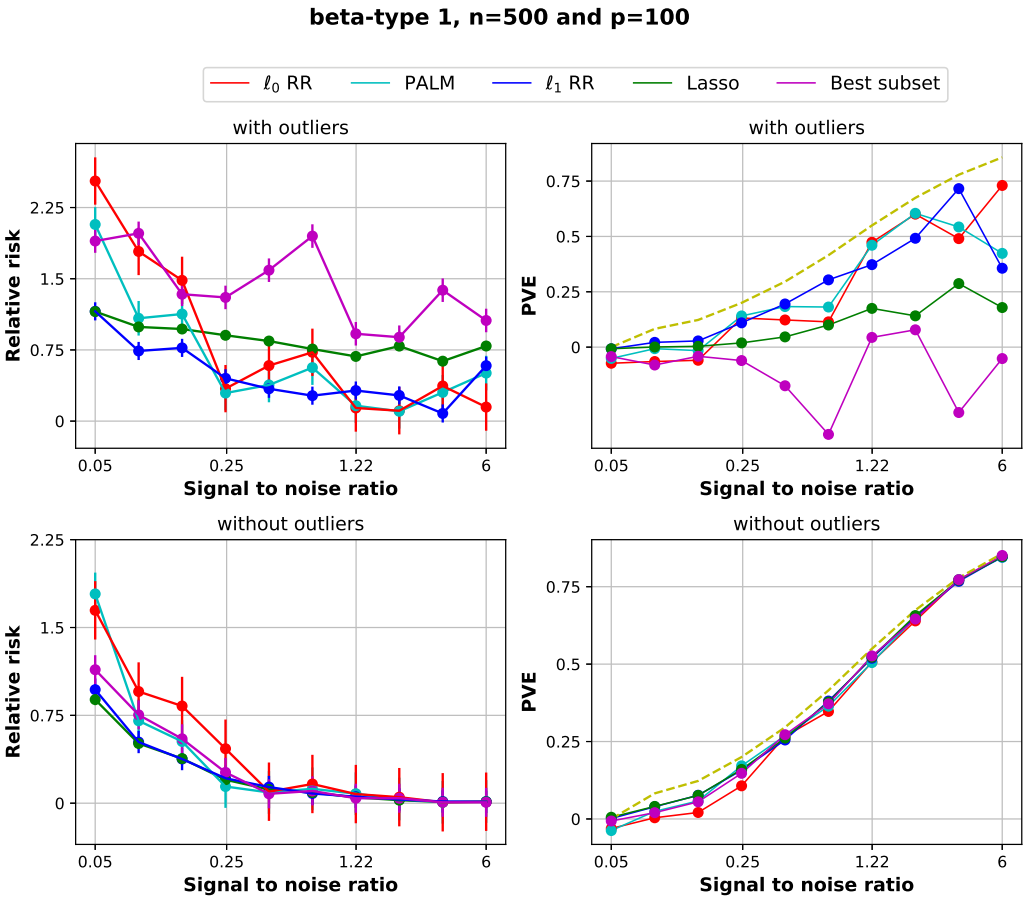

Figure 6: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 1 in the setting with* $n = 500$, $p = 100$, *and* $s = 5$ *with and without outliers (top panel and bottom panel respectively).*

5 Real Data Sets

The performances of all methods have been compared on real data sets. To this end we have used 7 data sets presented in Table 1. The different methods have been compared on all these data sets according to the following setup:

- The response vector y and the columns of the matrix X have been standardized to have zero mean and unit standard deviation;
- Two 5-fold cross-validation loops have been implemented. The inner one has been used to give a relevant choice for the hyper-parameters. The outer one has been used to estimate the average mean squared error MSE;

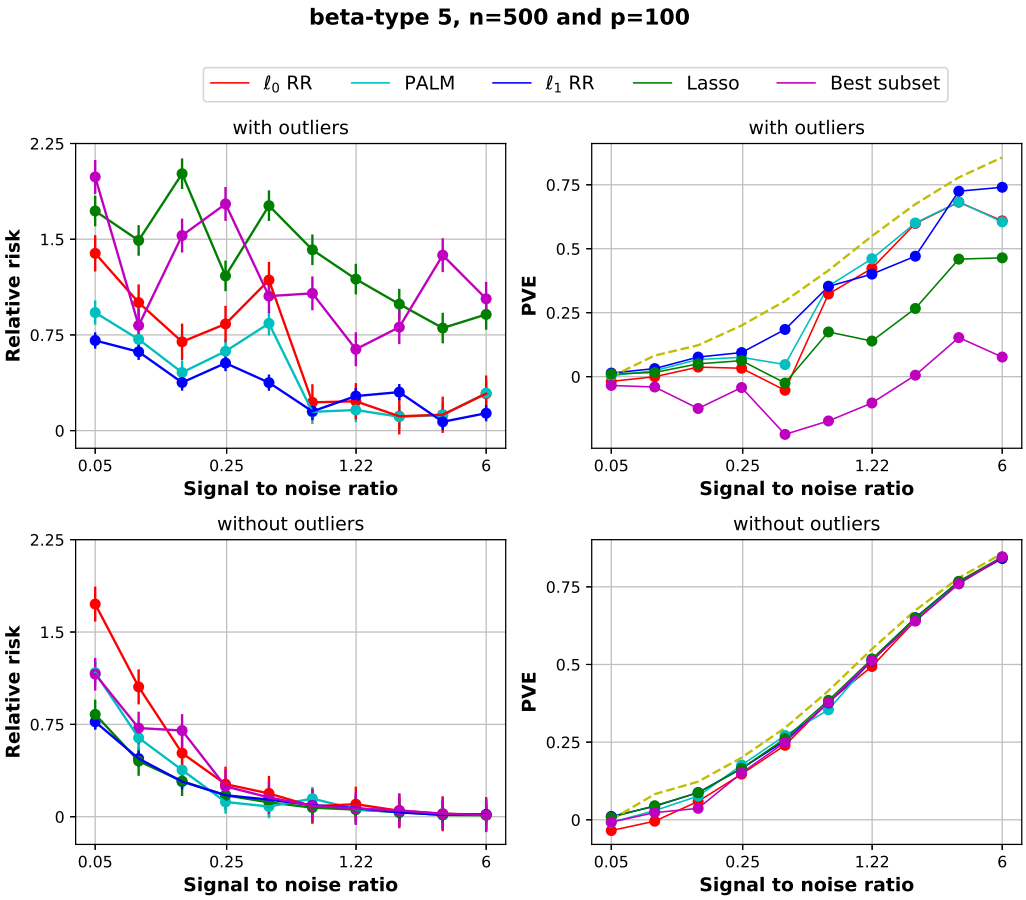

Figure 7: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 2 in the setting with $n = 500$, $p = 100$, and $s = 5$ with and without outliers (top panel and bottom panel respectively).*

- As for synthetic data sets, we run PALM for k_v ranging from 1 to p, and k_o ranging from 0 to 10% with a step seize of 2.5%, and pick the solution with smallest cross-validation error. This obtained solution is used to set the values of M_v and M_o and as a warm start for the ℓ_0 robust regression algorithm as well;
- The hyper-parameter λ of the lasso was tuned over 100 values as per the default in `glmnet`;
- The ℓ_1 robust regression algorithm was tuned over 5 values of λ (as for the synthetic data sets) and over 40 values of γ varying from 0 to 2000 with a step size of 50. We remarked that, for the normalized and standardized data set considered, it's enough to bound $||\tau||_1$ by 2000;
- Outliers were generated by replacing 5% of the response vector values y_i by $y_i + 2(\max(y) - \min(y))$ that is a constant value set to the range of the response variable in the training set;

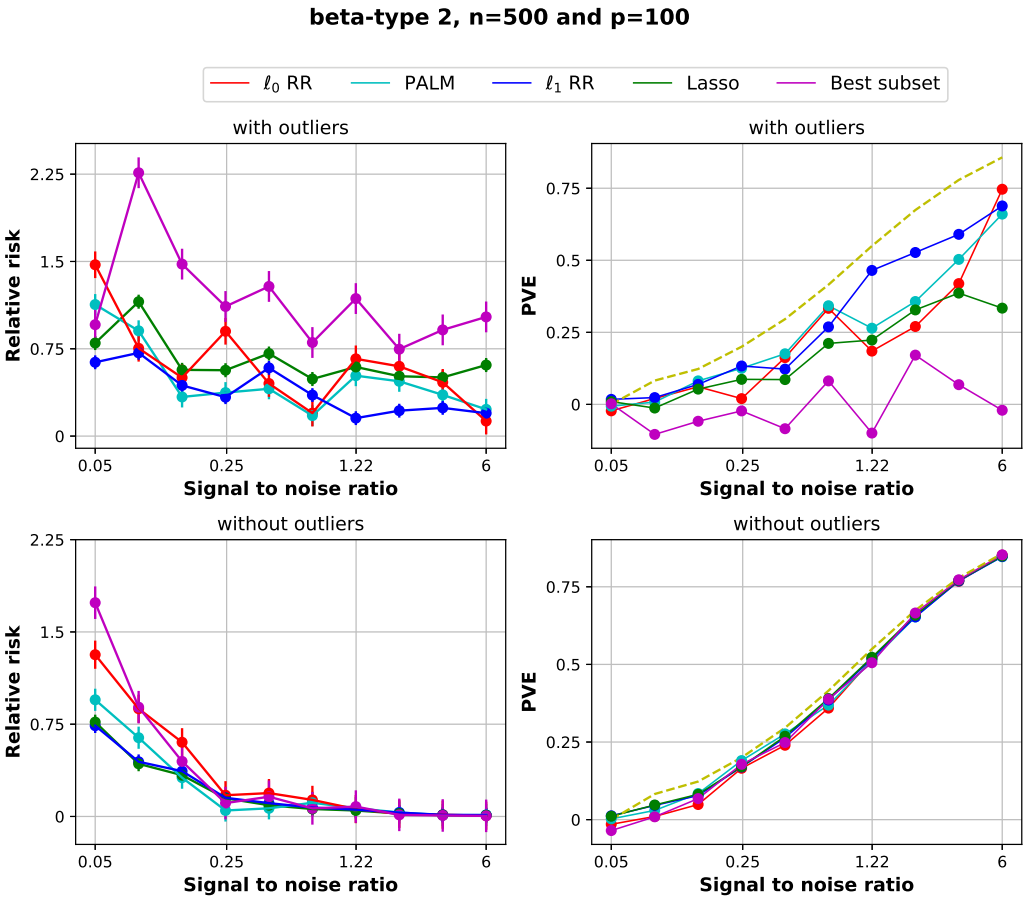

Figure 8: *Relative risk (left panel) and proportion of variance explained (right panel) functions of SNR, for beta-type 5 in the setting with* $n = 500$, $p = 100$, *and* $s = 5$ *with and without outliers (top panel and bottom panel respectively).*

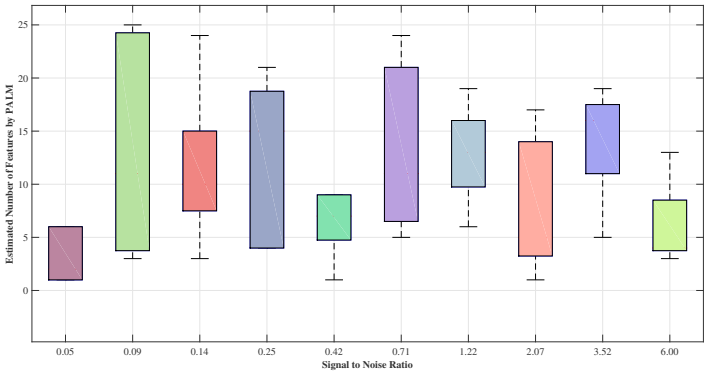

Figure 9: *Summary of used datasets.*

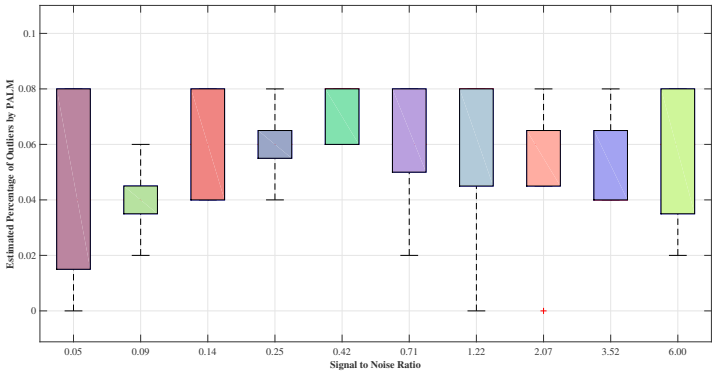

Figure 10: *Percentage of outliers over estimation by PALM. The percentage of outliers in the data set is 5%.*

Each experiment is repeated 3 times. Tables 2 and 3 report the average of the results and the standard deviation in parentheses for the raw data.

Table 1: *Periods and sites extracted from clear archaeological contexts with radiocarbon determinations.*

Name of the dataset	*number of instances n*	*number of attributes p*	*Origin*
Body Fat	252	15	`lib.stat.cmu.edu`
Concrete Compressive Strength	1030	9	UCI
Concrete Slump Test	103	10	UCI
Real Estate Valuation	414	7	UCI
Diabetes	442	10	`stat.ncsu.edu`
Boston Housing	489	3	Web[1]
Auto Mpg	398	8	UCI

Table 2: *Cross-validation MSE rates (standard deviations) of the best subset, lasso, PALM, ℓ_0 robust regression (ℓ_0 RR) and ℓ_1 robust regression (ℓ_1 RR) on 7 real datasets.*

	Best subset	Lasso	Palm	ℓ_0 RR	ℓ_1 RR
Body Fat	**2.2797** ($7.2e^{-5}$)	4.2644 ($1.5e^{-4}$)	2.5958 ($5.2e^{-5}$)	2.6270 ($4.77e^{-5}$)	4.5008 ($6.2e^{-5}$)
Concrete Compressive Strength	**0.3588** (0.018)	0.3602 (0.019)	0.3692 ($4.2e^{-4}$)	0.3693 ($3.5e^{-4}$)	0.3603 (0.015)
Slump Test	0.0880 (0.008)	**0.0863** (0.012)	0.0864 (0.011)	0.0880 (0.008)	0.0869 (0.010)
Real Estate Valuation	0.2994 (0.024)	**0.2924** (0.036)	0.3010 (0.026)	0.2992 (0.026)	0.2950 (0.033)
Diabetes	0.3917 (0.037)	0.3914 (0.038)	0.3889 (0.028)	**0.3888**(0.038)	0.3952 (0.039)
Boston Housing	0.2460 (0.007)	0.2460 (0.007)	0.2446 (0.008)	**0.2440** (0.009)	0.2448 (0.006)
Auto Mpg	0.1469 (0.002)	**0.1458** (0.005)	0.1523 (0.007)	0.1516 (0.007)	0.1478 (0.008)

An important caveat to emphasize upfront is that the ℓ_0 robust regression algorithm was given 10 minutes time limit per problem instance per subset size. This practical restriction may have caused this algorithm to underperform in some cases. For the best subset selection problem, the time limit was set to 2 minutes. We note that the optimality was certified for almost every case in less than two minutes. In the absence of outliers, results in Table 2 show that there is no clear winner. It is remarkable that all methods

performed quite similarly, with a little advantage of using the lasso. In the presence of outliers, results in Table 3 show the dominance of the robust regression algorithms used over the best subset selection and the lasso. The ℓ_0 robust regression performed better than the other methods.

Table 3: *Cross-validation MSE rates (standard deviations) of the best subset, lasso, PALM, ℓ_0 robust regression (ℓ_0 RR) and ℓ_1 robust regression (ℓ_1 RR) on 7 real datasets corrupted by 5% of outliers in the initial response vector y.*

	Best subset	Lasso	Palm	ℓ_0 RR	ℓ_1 RR
Body Fat	0.3923 (0.023)	0.4039 (0.034)	**0.3679** (0.024)	0.3764 (0.009)	0.3882 (0.023)
Concrete compressive strength	0.5891 (0.063)	0.5877 (0.059)	0.5843 (0.070)	**0.5842** (0.071)	0.5857 (0.755)
Slump test	0.2749 (0.186)	0.2463 (0.128)	0.1110 (0.022)	**0.0958** (0.012)	0.1039 (0.018)
Real estate valuation	0.6581 (0.131)	0.6680 (0.146)	0.6587 (0.137)	**0.6580** (0.138)	0.6688 (0.147)
Diabetes	0.5087 (0.015)	0.5002 (0.011)	0.5012 (0.009)	0.5009 (0.011)	**0.4923** (0.014)
Boston housing	0.5408 (0.240)	0.5293 (0.231)	0.5425 (0.241)	0.5441 (0.241)	**0.5235** (0.225)
Auto mpg	0.5498 (0.139)	0.5596 (0.128)	0.5406 (0.160)	0.5406 (0.160)	**0.5370** (0.163)

6 Conclusion

In this paper we propose a method for linear regression which solves the underlying optimization problem that handles both variable selection and outlier detection. We formulate the problem as a mixed-integer optimization problem and present a fast alternating minimization algorithm to find local minima. Furthermore, we present an empirical comparison between this method and its ℓ_1 relaxation on both synthetic and real data. We have found that neither the ℓ_0 norm problem nor its ℓ_1 relaxation dominates the other. Our recommendation is to use the ℓ_0 norm problem for large SNR while ℓ_1 relaxation is preferred when SNR is small. While the ℓ_0 approach is considered to be intractable, especially, for high dimensional regimes, one can propose to use screening rules helping in accelerating the solvers. Moreover, we have shown that if the true number of features and percentage of outliers are well estimated, the speed of convergence to the global minimum decreases significantly. Dealing with data sets of high dimensionality is the main limitation of the proposed MIO approach because of the high computational cost. However, we suggest to use the PALM algorithm in the high-dimensional case since it provides high quality solutions in a short time.

References

Alfons, A., Croux, C. and Gelper, S. et al. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7, 226–248.

Bertsimas, D., King, A. and Mazumder, R. (2015). Best subset selection via a modern optimization lens. *Annals of Statistics*, 47, 2324–2354.

Bolte, J., Sabach, S. and Teboulle, M. (2014). Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146, 459–494.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I. and Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30, 891–927.

Chen, Y., Caramanis, C. and Mannor, S. (2013). Robust sparse regression under adversarial corruption. In *International Conference on Machine Learning*, pp. 774–782.

Dalalyan, A. S. and Thompson, P. (2019). Outlier-robust estimation of a sparse linear model using ℓ_1-penalized huber's m-estimator. *arXiv preprint arXiv:1904.06288*.

Giloni, A. and Padberg, M. (2002). Least trimmed squares regression, least median squares regression, and mathematical programming. *Mathematical and Computer Modelling*, 35, 1043–1060.

Hastie, T., Tibshirani, R. and Tibshirani, R. J. (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, 22, 85–126.

Miller, A. (2002). *Subset selection in regression*. CRC Press.

Miyashiro, R. and Takano, Y. (2015). Subset selection by mallows: A mixed integer programming approach. *Expert Systems with Applications*, 42, 325–331.

Nguyen, N. H. and Tran, T. D. (2013). Robust lasso with missing and grossly corrupted observations. *IEEE transactions on information theory*, 59, 2036–2058.

Öllerer, V., Alfons, A. and Croux, C. (2016). The shooting s-estimator for robust regression. *Computational Statistics*, 31, 829–844.

Parikh, N. and Boyd, S. P. (2014). Proximal algorithms. *Foundations and Trends in optimization*, 1, 127–239.

Rousseeuw, P. J. and Hubert, M. (2018). Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8, e1236.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*, Volume 589. John Wiley & Sons.

She, Y. and Owen, A. B. (2011). Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106, 626–639.

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

Tibshirani, R., Wainwright, M. and Hastie, T. (2015). *Statistical Learning with Sparsity: the Lasso and Generalizations*. Chapman and Hall/CRC.

Wang, H., Li, G. and Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25, 347–355.

Yang, M., Xu, L., White, M., Schuurmans, D. and Yu, Y.-l. (2010). Relaxed clipping: A global training method for robust regression and classification. In *Advances in neural information processing systems*, pp. 2532–2540.