Outlier detection in regression: conic quadratic formulations *

Andrés Gómez† José Neto‡

June 2023

Abstract

In many applications, when building linear regression models, it is important to account for the presence of outliers, i.e., corrupted input data points. Such problems can be formulated as mixed-integer optimization problems involving cubic terms, each given by the product of a binary variable and a quadratic term of the continuous variables. Existing approaches in the literature, typically relying on the linearization of the cubic terms using big-M constraints, suffer from weak relaxation and poor performance in practice. In this work we derive stronger second-order conic relaxations that do not involve big-M constraints. Our computational experiments indicate that the proposed formulations are several orders-of-magnitude faster than existing big-M formulations in the literature for this problem.

1 Introduction

Several statistical and machine learning problems can be formulated as optimization problems of the form

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z}} \ & \sum_{i=1}^{m} (y_i - \boldsymbol{a_i}^\top \boldsymbol{x})^2 (1 - z_i) \\ \text{s.t. } & (\boldsymbol{x}, \boldsymbol{z}) \in F \subseteq \mathbb{R}^n \times \{0,1\}^m, \end{aligned} \tag{1}$$

where $(\boldsymbol{a_i}, y_i) \in \mathbb{R}^{n+1}$ for all $i \in \{1, \dots, m\}$ are given data and F is the feasible region. Problem (1) includes the *least trimmed squares* as a special case, which is a focus of this paper and discussed at length in §1.1, but also includes regression

*This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH. A Gómez is supported by grant FA9550-22-1-0369 of the US Air Force Office of Scientific Research.

†Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California, CA 90089 (gomezand@usc.edu).

‡SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France (jose.neto@telecom-sudparis.eu).

trees [17] (where $1 - z_i = 1$ indicates that a given datapoint is routed to a given leaf), regression problems with mismatched data [38] (where variables $\boldsymbol{z}$ indicate the datapoint/response pairs) and k-means [33] (where variables $\boldsymbol{z}$ represent assignment of datapoints to clusters). We point out that few or no mixed-integer optimization (MIO) approaches exist in the literature for (1), as the problems are notoriously hard to solve to optimality, and heuristics are preferred in practice.

The hardness of problem (1) is due to weak relaxations such as standard big-M relaxations, producing trivial lower bounds of 0 and gaps of 100%. The purpose of this work is thus to derive stronger relaxations of (1), paving the way for efficient exact methods via MIO.

1.1 Robust estimators and least trimmed squares

Most statistical methods fail if the input data is corrupted by so-called *outliers*. The latter correspond to erroneous input data points resulting, e.g., from measurement, transmission, recording errors or exceptional phenomena. Consider linear regression models, described by observations $\{(\boldsymbol{a_i}, y_i)\}_{i=1}^m$ where $\boldsymbol{a_i} \in \mathbb{R}^n$ are the features and y_i is the response associated with datapoint i. The classical ordinary least squares (OLS) estimator, defined as the minimizer of the optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^m (y_i - \boldsymbol{a_i}^\top \boldsymbol{x})^2 = \min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \tag{OLS}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is the matrix with rows given by $\{\boldsymbol{a_i}\}_{i=1}^m$, is known to be sensitive to spurious perturbations of the data. Two robust modifications of (OLS) are commonly used in practice. The first one calls for the addition of an additional regularization term, resulting in the *least squares with Tikhonov regularization* problem. Specifically, given a suitable matrix $\boldsymbol{T}$ (typically taken as the identity), the estimator is the optimal solution of

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^m (y_i - \boldsymbol{a_i}^\top \boldsymbol{x})^2 + \lambda\|\boldsymbol{T}\boldsymbol{x}\|_2^2, \tag{LS+L2}$$

which is robust against small perturbations of the data [18]. The second approach calls for replacing the least squares loss with the absolute value of the residuals, resulting in *least absolute deviations* (LAD) problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^m \left|y_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right|. \tag{LAD}$$

Estimator (LAD), which generalizes the median to multivariate regression, is preferred to (OLS) in settings with outliers.

Despite their popularity, (LS+L2) and (LAD) are known to be vulnerable to outliers. Robust estimators are often measured according to the breakdown

point [24] – the smallest proportion of contaminated data that can cause the estimator to take arbitrarily large aberrant values. Clearly, estimators (OLS) and (LS+L2) have an unfavorable breakdown point of 0%: a single spurious observation with $\boldsymbol{a_i} = \boldsymbol{e_j}$, where $\boldsymbol{e_j}$ is the j-th standard basis vector, and $y_i \to \pm\infty$ will produce solutions where x_j takes arbitrarily bad values. M-estimators [29, 30], which include as special cases (LAD) and regression with respect to the Huber loss, also have breakdown point of 0% [47].

Robust estimators with better breakdown point include the *least median of squares* (LMS) [43, 45] which minimizes the median of the squared residuals. The *least quantile of squares* (LQS) [11] approach generalizes the latter by minimizing the q-th order statistic, i.e., the q-th smallest residual in absolute value for some given integer $q \leq m$. The *least trimmed squares problem* (LTS) [43, 45], consists in minimizing, for some $h \in \mathbb{Z}$, the sum of the smallest h residual squares. Specifically, letting $r_i(\boldsymbol{x}) = |y_i - \boldsymbol{a_i}^\top \boldsymbol{x}|$ be the i-th residual, and letting $\left|r_{(1)}\left(\boldsymbol{x}\right)\right| \leq \left|r_{(2)}\left(\boldsymbol{x}\right)\right| \leq \ldots \leq \left|r_{(m)}\left(\boldsymbol{x}\right)\right|$ be the residuals sorted in nondecreasing magnitude order, the LTS estimator is the optimal solution of

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^{h} r_{(i)}\left(\boldsymbol{x}\right)^2 + \lambda\|\boldsymbol{T}\boldsymbol{x}\|_2^2. \tag{LTS+L2}$$

Note that for $h = m$, (LTS+L2) corresponds to (LS+L2). Intuitively, for $h \leq m-1$, the datapoints corresponding to the $m-h$ largest residuals are observations flagged as outliers and discarded prior to using (LS+L2) to fit a model on the remaining data. The original LTS estimator had $\lambda = 0$, but we consider here the version with additional ℓ_2 regularization used in [31], where the additional regularization helps counteracting strong collinearities between features and improves performance in low signal-to-noise regimes.

The LMS and LTS estimators achieve an optimal breakdown point of 50% [43, 45]. While LMS was more popular originally, as it is less difficult to compute, Rousseeuw and Van Driessen [46] argue that "the LMS estimator should be replaced by the LTS estimator" due to several desirable properties, including smoothness and statistical efficiency. Unfortunately, computing the LTS estimator is NP-hard [9, 8] and even hard to approximate [40].

For the most part, problem (LTS+L2) is solved using heuristics. In particular, methods which alternate between fitting regression coefficients given a fixed set of h non-outlier observations and determining new outliers given fixed regression coefficients $\bar{\boldsymbol{x}}$ are popular in the literature [25, 46]. Solution methods based on solving least trimmed squares with similar iterative approaches have also been proposed in the context of mixed linear regression with corruptions and more general problems, e.g. [48, 49] and references therein. Under some specific assumptions on the model, convergence results to an optimal solution have been established for such algorithmic schemes [13]. However, in general, they do not provide guarantees and the quality of the resulting estimators can be poor.

Agulló [1] proposed a branch and bound algorithm to solve (LTS+L2) to optimality, which is shown to be fast in instances with $m \leq 30$, but struggles in larger instances. A first MIO formulation for (LTS+L2) was proposed in [20],

although the authors observe that the resulting optimization problem is difficult to solve and do not provide computations. To the best of our knowledge, the first implementation of a MIO algorithm for (LTS+L2) was done in [56], based on a formulation using big-M constraints, where the authors report solution times of two seconds for instances with $m = 25$ and also comment on larger computational times for larger instances. In a subsequent work by the same research group [57], the authors report solution times in seconds for problems with $n = 2$ and $m \leq 50$, and in minutes for problems with $100 \leq m \leq 500$, although all computations are performed on synthetic data. In a recent paper, [31] propose another big-M formulation for a generalization of (LTS+L2) (where sparisty is also imposed on regression variables $\boldsymbol{x}$), and report computational times in minutes for synthetic instances with n and m in the low hundreds. We discuss these MIO approaches further in §2. Finally, we point that exact big-M based MIO algorithms and continuous optimization heuristics were proposed in [11] for the related LMS problem: the authors report that MIO methods are dramatically outperformed by the continuous optimization approaches, with the objective value of MIO solutions being up to 400x worse than the objective of heuristic solutions (unless the heuristics solutions are used as a warm-start).

1.2 Contributions, outline and notation

In this work, we introduce strong, big-M free, conic quadratic reformulations for (LTS+L2) –and, more generally, problems of the form (1). Extensive computational experiments on diverse families of instances (both synthetic and real) clearly point out strong improvements over current state-of-the-art approaches. In particular, the proposed formulations results in orders-of-magnitude improvements over existing big-M formulations in our computations. We refrain from providing an estimate of the scalability of the approach: we show instances with $(n, m) = (20, 500)$ that are solved in 10 seconds, and instances with $(n, m) = (4, 50)$ that cannot be solved within a time limit of 10 minutes. Indeed, for MIO approaches, the effectiveness of the approach depends on more factors than simply the size of the instance, including the number $m-h$ of observations to be discarded, the regularization parameter λ, and the overall structure of the dataset (with synthetic instances being considerably easier than real ones).

The paper is organized as follows. We close this section with some notation. In §2 we review the literature on MIO approaches for linear regression problems. Convexification results related to sets originating from (1) are presented in §3. The convexifications are used to derive conic quadratic reformulations of (LTS+L2) in §4. The experimental framework and computational results are presented in §5 and we conclude the paper in §6.

Notation. For any positive integer n, let $[n]$ stand for the set $\{1, 2, \ldots, n\}$. The vectors and matrices are represented with **bold** characters. The all-zero and all-one vectors and matrices (with appropriate dimensions) are represented by $\mathbf{0}$ and $\mathbf{1}$ respectively. The i-th unit vector is represented by $\boldsymbol{e_i}$. The notation $\boldsymbol{I}$ stands for the identity matrix. Given a vector $\boldsymbol{d} \in \mathbb{R}^n$, we let $\text{Diag}(\boldsymbol{d}) \in \mathbb{R}^{n\times n}$ denote the diagonal matrix with elements $\text{Diag}(\boldsymbol{d})_{ii} = d_i$. Given a square matrix

$\boldsymbol{Q}$, we let $\boldsymbol{Q}^\dagger$ denote the pseudoinverse of $\boldsymbol{Q}$.

2 Review of MIO methods for outlier detection

There has been a recent trend of using mathematical optimization techniques to tackle hard problems arising in the context of linear regression. In particular, there is a stream of research focused on the best subset selection problem [4, 5, 7, 10, 12, 26, 28, 27, 37, 54, 14, 15, 53, 16], in which at most k of the regression variables in (LS+L2) can take non-zero values. Variants of best subset selection, in which information criteria are used to determine the number of non-zero variables, have also been considered in the literature [32, 42, 39, 22]. Related models have also been used to tackle inference problems with graphical models and sparsity [36, 34, 6]. We point out that most of the approaches for sparse regression are based on improving continuous relaxations by exploiting a ridge regularization term $\lambda\|\boldsymbol{x}\|_2^2$ through the perspective reformulation [19, 23]. As we show in this paper, the Tikhonov regularization $\lambda\|\boldsymbol{T}\boldsymbol{x}\|_2^2$ is also fundamental for improving relaxations for (LTS+L2).

Despite the plethora of MIO approaches for sparse regression, there is a dearth of similar methods for regression problems with outliers. Indeed, problems such as (LTS+L2) appear to be fundamentally more difficult than sparse regression problems. Observe that problem (LTS+L2) admits the natural mixed-integer cubic formulation [20]

$$\min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{z}\in\{0,1\}^m} \sum_{i=1}^m \left(y_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 (1-z_i) + \lambda\|\boldsymbol{T}\boldsymbol{x}\|_2^2 \text{ s.t. } \mathbf{1}^\top \boldsymbol{z} \leq m-h, \tag{2}$$

where $z_i = 1$ if datapoint i is flagged as an outlier and discarded, and $z_i = 0$ otherwise. Note that (2) is a special case of (1), where F is given by a cardinality constraint. Formulation (2) cannot be effectively used with most MIO software. Indeed, its natural continuous relaxation, obtained by relaxing the binary constraints to bound constraints $\mathbf{0} \leq \boldsymbol{z} \leq \mathbf{1}$, is non-convex. To circumvent this issue, Zioutas et al. [56, 57] reformulated (2) as the convex quadratic mixed integer optimization problem

$$\min_{\boldsymbol{u},\boldsymbol{x},\boldsymbol{z}} \sum_{i=1}^m u_i^2 + \lambda\|\boldsymbol{T}\boldsymbol{x}\|_2^2 \tag{3a}$$

$$\text{s.t.} -y_i + \boldsymbol{a_i}^\top \boldsymbol{x} \leq u_i + z_i M \qquad \forall i \in [m] \tag{3b}$$

$$y_i - \boldsymbol{a_i}^\top \boldsymbol{x} \leq u_i + z_i M \qquad \forall i \in [m] \tag{3c}$$

$$\mathbf{1}^\top \boldsymbol{z} \leq m-h \tag{3d}$$

$$\boldsymbol{u} \in \mathbb{R}_+^m, \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in \{0,1\}^m \tag{3e}$$

where M is a sufficiently large fixed constant and u_i represents the absolute value of the i-th residual. Indeed, in any optimal solution $(\boldsymbol{u^*}, \boldsymbol{x^*}, \boldsymbol{z^*})$ of (3), having $z_i^* = 1$ (resp. $z_i^* = 0$) implies $u_i^* = 0$ (resp. $u_i^* = \left|y_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right|$), i.e. the objective value is sum of the squared residuals of the non-outlier datapoints.

While formulation (3) can be directly used with most MIO solvers, the natural continuous relaxation is trivial. Indeed, regardless of the data $(\boldsymbol{A}, \boldsymbol{y}, \boldsymbol{T})$, an optimal solution of the continuous relaxation is given by $\boldsymbol{u}^* = \boldsymbol{0}$, $\boldsymbol{x}^* = \boldsymbol{0}$ and $\boldsymbol{z}^* = ((m-h)/m)\,\boldsymbol{1}$. The objective value of this relaxation is thus equal to the trivial lower bound of 0 (resulting in a 100% optimality gap), which leads to large branch-and-bound trees as solvers cannot effectively prune the search space. Moreover, the solutions of the continuous relaxations are essentially uninformative, thus MIO solvers –which rely on these to produce feasible solutions and inform branching decisions– struggle to tackle problem (3).

In fact, as we show in §3, any relaxation based on a convex reformulation of the individual cubic terms $\left(y_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 (1 - z_i)$ necessarily results in trivial bounds and solutions. We point out that this phenomenon sets apart regression problems with outliers from sparse regression problems: the continuous relaxations of the natural big-M formulations of sparse regression problems (e.g., see [10]) is equivalent to the least squares problem (LS+L2), producing non-trivial bounds and solutions. We conjecture that the difficulty to produce a "reasonable" convex relaxation of (2) is the reason why few MIO approaches exist for regression problems with outliers. A notable exception is [21] , which proposes strong conic quadratic formulations for outlier detection with time series data. However, the methodology proposed in that paper is tailored to time series data and cannot be generalized to problem (LTS+L2).

3 Convexification results

In this section we investigate the convex hull of sets related to terms arising in the formulation of problems such as (LTS+L2). To motivate our approach, let us first study the convex hull of the set

$$Y_c = \left\{(\boldsymbol{x}, z, t) \in \mathbb{R}^n \times \{0,1\} \times \mathbb{R} \colon t \geq \left(c - \boldsymbol{a}^\top \boldsymbol{x}\right)^2 (1 - z)\right\}$$

where $c \in \mathbb{R}$ is a scalar. Y_c may be interpreted as the mixed-integer epigraph of the error function associated with a single datapoint. As Proposition 1 below shows, any closed convex relaxation of Y_c is trivial. In other words, any formulation of (1) based only on convex reformulations of each individual cubic term will result in 100% gaps and non-informative relaxations.

Proposition 1. *The closure of the convex hull of Y_c is given by*

$$cl\ conv(Y_c) = \mathbb{R}^n \times [0,1] \times \mathbb{R}_+.$$

Proof. Consider any point $(\bar{\boldsymbol{x}}, \bar{z}, \bar{t}) \in \mathbb{R}^n \times [0,1] \times \mathbb{R}_+$ with $0 < \bar{z} < 1$. Observe that

$$(\bar{\boldsymbol{x}}, \bar{z}, \bar{t}) = \bar{z}\left(\frac{\bar{\boldsymbol{x}}}{\bar{z}} - \frac{1-\bar{z}}{\bar{z}}\frac{c \cdot \boldsymbol{a}}{\|\boldsymbol{a}\|_2^2}, 1, 0\right) + (1-\bar{z})\left(\frac{c \cdot \boldsymbol{a}}{\|\boldsymbol{a}\|_2^2}, 0, \frac{\bar{t}}{1-\bar{z}}\right),$$

where both $\left(\frac{\bar{\boldsymbol{x}}}{\bar{z}} - \frac{1-\bar{z}}{\bar{z}}\frac{c\cdot\boldsymbol{a}}{\|\boldsymbol{a}\|_2^2}, 1, 0\right) \in Y_c$ and $\left(\frac{c\cdot\boldsymbol{a}}{\|\boldsymbol{a}\|_2^2}, 0, \frac{\bar{t}}{1-\bar{z}}\right) \in Y_c$, and thus $(\bar{\boldsymbol{x}}, \bar{z}, \bar{t}) \in \operatorname{conv}(Y_c)$. Moreover, since $(\bar{\boldsymbol{x}}, 0, \bar{t}) = \lim_{z\to 0^+}(\bar{\boldsymbol{x}}, z, \bar{t})$, we find that $(\bar{\boldsymbol{x}}, \bar{z}, \bar{t}) \in \operatorname{cl}\operatorname{conv}(Y_c)$ even if $z = 0$. □

Thus, to derive stronger relaxations, it is necessary to study a more general set, capturing more structural information about the optimization problem. In particular, the formulations we propose to tackle problem (LTS+L2) are based on a study of the set

$$Y_{c,\boldsymbol{Q}} = \left\{(\boldsymbol{x}, z, t) \in \mathbb{R}^n \times \{0,1\} \times \mathbb{R} \colon t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(c - \boldsymbol{a}^\top \boldsymbol{x}\right)^2 (1-z)\right\} \quad (4)$$

where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ represents a symmetric and positive definite matrix. We provide hereafter descriptions of the convex hull of $Y_{c,\boldsymbol{Q}}$ in the original space of variables for the homogeneous case (i.e., when $c = 0$) and in an extended space in the non-homogeneous case ($c \neq 0$).

3.1 Convexification of $Y_{0,\boldsymbol{Q}}$

The convexification of set

$$Y_{0,\boldsymbol{Q}} = \left\{(\boldsymbol{x}, z, t) \in \mathbb{R}^n \times \{0,1\} \times \mathbb{R} \colon t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2 (1-z)\right\}$$

admits a relatively simple description in the original space of variables.

Proposition 2. *The closure of the convex hull of set $Y_{0,\boldsymbol{Q}}$ is*

$$\textit{cl conv}(Y_{0,\boldsymbol{Q}}) = \left\{(\boldsymbol{x}, z, t) \in \mathbb{R}^n \times [0,1] \times \mathbb{R} \colon t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \frac{(1-z)\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2}{1 + z\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2}\right\}.$$

Proof. Let T denote the set in the right-hand side of the equation in the statement of the proposition. We show next that: • T is convex, • T induces a relaxation of $Y_{0,\boldsymbol{Q}}$, and • optimization of a linear function over set T is equivalent to optimization over $Y_{0,\boldsymbol{Q}}$.

• **Convexity** We show convexity of T by establishing it is equivalent to the SDP-representable set given by constraints

$$0 \leq z \leq 1,\ \begin{pmatrix} \boldsymbol{W} & \boldsymbol{x} \\ \boldsymbol{x}^\top & t \end{pmatrix} \succeq 0,\ \boldsymbol{W} = \boldsymbol{Q^{-1}} - (1-z)\frac{\boldsymbol{Q^{-1}}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q^{-1}}}{1 + \|\boldsymbol{Q^{-1/2}}\boldsymbol{a}\|_2^2}.$$

Note that $\boldsymbol{W} \succ 0$ since for any $\boldsymbol{y} \neq \boldsymbol{0}$,

$$\boldsymbol{y}^\top \boldsymbol{W}\boldsymbol{y} \geq \boldsymbol{y}^\top \boldsymbol{Q^{-1}}\boldsymbol{y} - \frac{\left(\boldsymbol{a}^\top \boldsymbol{Q^{-1}}\boldsymbol{y}\right)^2}{1 + \left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2} \geq \left(\boldsymbol{y}^\top \boldsymbol{Q}^{-1}\boldsymbol{y}\right)\cdot\left(1 - \frac{\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2}{1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2}\right) > 0,$$

where the second inequality uses Cauchy-Schwarz inequality and the last one follows from $\boldsymbol{y} \neq \boldsymbol{0}$ and the definition of $\boldsymbol{Q^{-1}} \succ 0$. Since $\boldsymbol{W}$ is invertible, we

find by using the Schur complement [3] that $\begin{pmatrix} \boldsymbol{W} & \boldsymbol{x} \\ \boldsymbol{x}^\top & t \end{pmatrix} \succeq 0 \Leftrightarrow t \geq \boldsymbol{x}^\top \boldsymbol{W^{-1}} \boldsymbol{x}$, and using the Sherman Morrison formula [50, 51] we can establish that $\boldsymbol{W^{-1}} = \boldsymbol{Q} + \frac{1-z}{1+z\|\boldsymbol{Q^{-1/2}a}\|_2^2} \boldsymbol{a}\boldsymbol{a}^\top$.

• **Relaxation** Observe that if $z = 0$ then T reduces to the inequality $t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{a}^\top \boldsymbol{x})^2$, and if $z = 1$ then T reduces to $t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x}$. This is precisely the disjunction encoded by $Y_{0,\boldsymbol{Q}}$, hence T is indeed a relaxation.

• **Equivalence** Now, to prove $T \subseteq \text{cl conv}\,(Y_{0,\boldsymbol{Q}})$, let us consider the optimization of an arbitrary linear function over the sets $Y_{0,\boldsymbol{Q}}$ and T:

$$\min_{(\boldsymbol{x},z,t)\in Y_{0,\boldsymbol{Q}}} \boldsymbol{\alpha}^\top \boldsymbol{x} + \beta z + \gamma t \tag{5}$$

$$\min_{(\boldsymbol{x},z,t)\in T} \boldsymbol{\alpha}^\top \boldsymbol{x} + \beta z + \gamma t \tag{6}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\beta \in \mathbb{R}$ and $\gamma \in \mathbb{R}$. Obviously if (6) has an optimal solution $(\boldsymbol{x^*}, z^*, t^*)$ with $z^* \in \{0, 1\}$, then it is also an optimal solution for (5). We then show that whenever (6) admits an optimal solution, there exists one with z binary. And if no optimal solution exists, then both problems (5)-(6) are unbounded.

We can assume that $\gamma > 0$ since (6) trivially has a binary solution if $\gamma = 0$ and $\boldsymbol{\alpha} = \boldsymbol{0}$, or both problems are unbounded (for any other combination of parameters with $\gamma \leq 0$). Moreover, by scaling, we can suppose that $\gamma = 1$. Then, assume that (6) has an optimal solution $(\boldsymbol{x}^*, z^*, t^*)$ with $0 < z^* < 1$. The point $(\boldsymbol{x}^*, z^*)$ is an optimal solution of

$$\min_{(\boldsymbol{x},z)\in\mathbb{R}^n\times[0,1]} q\,(\boldsymbol{x}, z) \tag{7}$$

with

$$q(\boldsymbol{x}, z) = \boldsymbol{\alpha}^\top \boldsymbol{x} + \beta z + \left\|\boldsymbol{Q}^{1/2}\boldsymbol{x}\right\|_2^2 + \frac{(1-z)\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2}{1 + z\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2}. \tag{8}$$

Fixing z in (8) and using the first order optimality conditions, we deduce the following expression of an optimal solution $\boldsymbol{x}(z)$ of $\min_{\boldsymbol{x}\in\mathbb{R}^n} q\,(\boldsymbol{x}, z)$:

$$\boldsymbol{x}(z) = -\frac{1}{2}\boldsymbol{Q}^{-1}\boldsymbol{\alpha} + \frac{1-z}{2\left(1 + \left\|\boldsymbol{Q}^{-1/2a}\right\|_2^2\right)}\boldsymbol{Q}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q}^{-1}\boldsymbol{\alpha}. \tag{9}$$

Thus, problem (7) reduces to $\min_{z\in[0,1]} q\,(\boldsymbol{x}(z), z)$. Substituting $\boldsymbol{x}(z)$ by its expression (9) in (8), we obtain that $q\,(\boldsymbol{x}(z), z)$ is a linear function of z. To be more precise, after computations, we get the following expression.

$$q\,(\boldsymbol{x}(z), z) = \beta z - \frac{1}{4}\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{\alpha}\right\|_2^2 + \frac{\left(\boldsymbol{a}^\top\boldsymbol{Q}^{-1}\boldsymbol{\alpha}\right)^2}{4\left(1 + \left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2\right)}(1-z). \tag{10}$$

Thus, (7) admits an optimal solution with $z \in \{0, 1\}$, concluding the proof. □

Intuitively, since terms $(1-z)(\boldsymbol{a}^\top \boldsymbol{x})^2$ do not admit a good convex reformulation (Proposition 1), the key is to instead use the *non-convex* reformulation $r(\boldsymbol{x}) = \frac{(1-z)\left(\boldsymbol{a}^\top \boldsymbol{x}\right)^2}{1+z\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2}$. To illustrate, consider the case with $n = 1$ and $a = 1$, that is,

$$Y_{0,\lambda} = \left\{(x,z,t) \in \mathbb{R} \times \{0,1\} \times \mathbb{R} \colon t \geq \lambda x^2 + x^2\,(1-z)\right\}, \text{ and}$$

$$\text{cl conv}\,(Y_{0,\lambda}) = \left\{(x,z,t) \in \mathbb{R} \times [0,1] \times \mathbb{R} \colon t \geq \lambda x^2 + \frac{(1-z)\,x^2}{1+z/\lambda}\right\},$$

where $\lambda > 0$ is a parameter that controls the magnitude of the quadratic term. Figure 1 (top) depicts the graphs of the convex envelopes $t = \lambda x^2 + \frac{(1-z)x^2}{1+z/\lambda}$ for various values of λ. Moreover, Figure 1 (bottom) depicts the graphs of the non-convex reformulation $r = \frac{(1-z)x^2}{1+z/\lambda}$ for the associated values of λ. Note that r can also be interpreted as the quantity added to the relaxation induced by big-M relaxations such as (3), which discard terms associated with $x^2(1-z)$ altogether. We observe that larger improvements over big-M relaxations are achieved for larger values of parameter λ.

3.2 Convexification of $Y_{c,\boldsymbol{Q}}$

We now consider the non-homogeneous case where $c \neq 0$. We could not establish a simple description of cl conv$(Y_{c,\boldsymbol{Q}})$ in the original space of variables. Moreover, while relaxations of $Y_{c,\boldsymbol{Q}}$ can be derived from Proposition 2 by writing $Y_{c,\boldsymbol{Q}} = \left\{(x_0,\boldsymbol{x},z,t) \in \mathbb{R}^{n+1} \times \{0,1\} \times \mathbb{R} \colon t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(cx_0 - \boldsymbol{a}^\top \boldsymbol{x}\right)^2 (1-z)\,, x_0 = 1\right\}$, we found in preliminary computations that the resulting convexifications (which do not account for constraint $x_0 = 1$) could be much weaker. Fortunately, as we show in this section, cl conv$(Y_{c,\boldsymbol{Q}})$ admits an easy representation with the introduction of an additional variable.

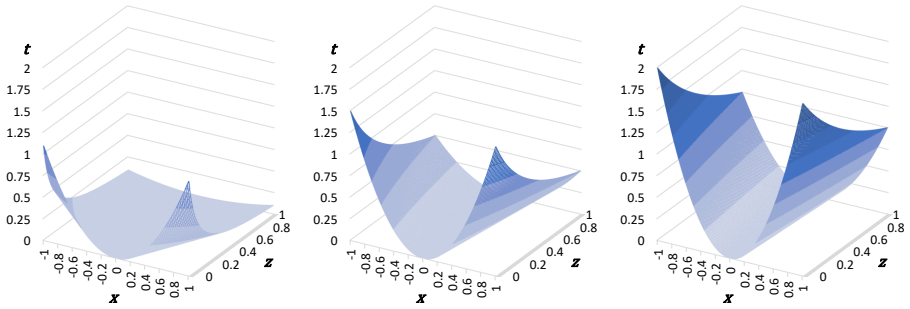
Observe that set $Y_{c,\boldsymbol{Q}}$ can be written as projection onto the $(\boldsymbol{x},z,t)$ space of

$$\hat{Y}_{c,\boldsymbol{Q}} = \Big\{\,(\boldsymbol{x},z,t,w) \in \mathbb{R}^n \times \{0,1\} \times \mathbb{R}^2 \colon t \geq \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(c + w - \boldsymbol{a}^\top \boldsymbol{x}\right)^2, \\ w(1-z) = 0\Big\}.$$

Indeed, if $z = 0$, then $w = 0$ and $Y_{c,\boldsymbol{Q}}$ and $\hat{Y}_{c,\boldsymbol{Q}}$ coincide. On the other hand, if $(\boldsymbol{x},1,t) \in Y_{c,\boldsymbol{Q}}$, then $(\boldsymbol{x},1,t,\boldsymbol{a}^\top \boldsymbol{x} - c) \in \hat{Y}_{c,\boldsymbol{Q}}$. We now characterize cl conv$(\hat{Y}_{c,\boldsymbol{Q}})$. Let $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ be any matrix such that $\boldsymbol{L}\boldsymbol{L}^\top = \left(\boldsymbol{Q} + \boldsymbol{a}\boldsymbol{a}^\top\right)^{-1}$, obtained for example from a Cholesky decomposition.

Theorem 1. *The closure of the convex hull of set $\hat{Y}_{c,\boldsymbol{Q}}$ is*

$$\textit{cl conv}\left(\hat{Y}_{c,\boldsymbol{Q}}\right) = \Bigg\{\,(\boldsymbol{x},z,t,w) \in \mathbb{R}^n \times [0,1] \times \mathbb{R}^2 \colon \\ t \geq c^2 + 2c(w - \boldsymbol{a}^\top \boldsymbol{x}) + \left\|\boldsymbol{L^{-1}}\left(\boldsymbol{x} - \frac{\boldsymbol{Q^{-1}a}}{1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} w\right)\right\|_2^2 + \frac{w^2}{\left(1 + \|\boldsymbol{Q^{-1/2}a}\|_2^2\right) z}\Bigg\}.$$



(a) $t = 0.1x^2 + \frac{(1-z)x^2}{1+z/0.1}$ (b) $t = 0.5x^2 + \frac{(1-z)x^2}{1+z/0.5}$ (c) $t = 1.0x^2 + \frac{(1-z)x^2}{1+z/1.0}$

(d) $r = \frac{(1-z)x^2}{1+z/0.1}$ (e) $r = \frac{(1-z)x^2}{1+z/0.5}$ (f) $r = \frac{(1-z)x^2}{1+z/1.0}$

Figure 1: Graphs of the convex envelopes $t = \lambda x^2 + \frac{(1-z)x^2}{1+z/\lambda}$ (top) and the non-convex reformulation $r = \frac{(1-z)x^2}{1+z/\lambda}$ (bottom) for $\lambda \in \{0.1, 0.5, 1.0\}$. Note that while the reformulation induced by r is non-convex, the convex envelope is due the strict convexity of term λx^2.

Proof. In the proof, first we compute cl conv $\left(\hat{Y}_{c,\boldsymbol{Q}}\right)$ in an SDP-representable extended formulation, then we simplify to a lower-dimensional SOCP-representable set, and finally we project out all additional variables.

SDP-representable formulation Observe that

$$\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \left(c + w - \boldsymbol{a}^\top \boldsymbol{x}\right)^2 = c^2 + 2c\left(w - \boldsymbol{a}^\top \boldsymbol{x}\right) + \begin{pmatrix}\boldsymbol{x}^\top & w\end{pmatrix} \boldsymbol{Q_1} \begin{pmatrix}\boldsymbol{x} \\ w\end{pmatrix} \tag{11}$$

with $\boldsymbol{Q_1} = \left(\begin{array}{c|c} \boldsymbol{Q}+\boldsymbol{a}\boldsymbol{a}^\top & -\boldsymbol{a} \\ \hline -\boldsymbol{a}^\top & 1 \end{array}\right)$. Define $\boldsymbol{Q_0} = \left(\begin{array}{c|c} \boldsymbol{Q}+\boldsymbol{a}\boldsymbol{a}^\top & \boldsymbol{0} \\ \hline \boldsymbol{0}^\top & 0 \end{array}\right)$. Then a description of cl conv $\left(\hat{Y}_{c,\boldsymbol{Q}}\right)$ in an extended formulation is [52]

$$\begin{aligned} \text{cl conv}\left(\hat{Y}_{c,\boldsymbol{Q}}\right) = \Big\{ (\boldsymbol{x}, z, t, w) \in \mathbb{R}^{n+3} \colon \exists \boldsymbol{W} \in \mathbb{R}^{(n+1)\times(n+1)}, \tau \in \mathbb{R} \text{ s.t.} \\ t \geq c^2 + 2c\left(w - \boldsymbol{a}^\top \boldsymbol{x}\right) + \tau \\ \begin{pmatrix} \tau & \boldsymbol{x}^\top & w \\ \boldsymbol{x} & & \boldsymbol{W} \\ w & & \end{pmatrix} \succeq 0 \\ (z, \boldsymbol{W}) \in \text{conv}(P) \Big\}, \end{aligned} \tag{12}$$

where $P = \left\{(0, \boldsymbol{Q_0^\dagger}), (1, \boldsymbol{Q_1^\dagger})\right\}$ and $\boldsymbol{Q_i^\dagger}$ denotes the pseudoinverse of $\boldsymbol{Q_i}$. Clearly, $\text{conv}(P) = \{(z, \boldsymbol{W}) \in [0,1] \times \mathbb{R}^{(n+1)\times(n+1)} : \boldsymbol{W} = (1-z)\boldsymbol{Q_0^\dagger} + z\boldsymbol{Q_1^\dagger}\}$.

SOCP-representable formulation Note that expressions of $\boldsymbol{Q_0^\dagger}$ and $\boldsymbol{Q_1^\dagger}$ can be easily computed [35]:

$$\boldsymbol{Q_0^\dagger} = \left(\begin{array}{c|c} \left(\boldsymbol{Q}+\boldsymbol{a}\boldsymbol{a}^\top\right)^{-1} & \boldsymbol{0} \\ \hline \boldsymbol{0}^\top & 0 \end{array}\right) = \left(\begin{array}{c|c} \boldsymbol{Q}^{-1} - \frac{\boldsymbol{Q}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q}^{-1}}{1+\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} & \boldsymbol{0} \\ \hline \boldsymbol{0}^\top & 0 \end{array}\right)$$

$$\boldsymbol{Q_1^\dagger} = \boldsymbol{Q_1^{-1}} = \left(\begin{array}{c|c} \boldsymbol{Q^{-1}} & \boldsymbol{Q}^{-1}\boldsymbol{a} \\ \hline \boldsymbol{a}^\top\boldsymbol{Q}^{-1} & 1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2 \end{array}\right).$$

Therefore, we find that constraint $\boldsymbol{W} = (1-z)\boldsymbol{Q_0^\dagger} + z\boldsymbol{Q_1^\dagger}$ simplifies to

$$\begin{aligned} \boldsymbol{W} &= \left(\begin{array}{c|c} \boldsymbol{Q}^{-1} - \frac{\boldsymbol{Q}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q}^{-1}}{1+\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} & \boldsymbol{0} \\ \hline \boldsymbol{0}^\top & 0 \end{array}\right) + z \left(\begin{array}{c|c} \frac{\boldsymbol{Q}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q}^{-1}}{1+\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} & \boldsymbol{Q}^{-1}\boldsymbol{a} \\ \hline \boldsymbol{a}^\top\boldsymbol{Q}^{-1} & 1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2 \end{array}\right) \\ &= \boldsymbol{U} + z\boldsymbol{v}\boldsymbol{v}^\top \end{aligned}$$

where

$$\boldsymbol{U} = \left(\begin{array}{c|c} \boldsymbol{Q}^{-1} - \frac{\boldsymbol{Q}^{-1}\boldsymbol{a}\boldsymbol{a}^\top\boldsymbol{Q}^{-1}}{1+\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} & \boldsymbol{0} \\ \hline \boldsymbol{0}^\top & 0 \end{array}\right) \text{ and } \boldsymbol{v} = \begin{pmatrix} \frac{\boldsymbol{Q}^{-1}\boldsymbol{a}}{\sqrt{1+\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2}} \\ \sqrt{1+\left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2} \end{pmatrix}.$$

Moreover, the system

$$\exists \boldsymbol{W} \in \mathbb{R}^{(n+1)\times(n+1)} \text{ s.t. } \begin{pmatrix} \tau & \boldsymbol{x}^\top & w \\ \boldsymbol{x} & & \boldsymbol{W} \\ w & & \end{pmatrix} \succeq 0,\ \boldsymbol{W} = \boldsymbol{U} + z\boldsymbol{v}\boldsymbol{v}^\top \tag{13}$$

can be reformulated as an SOCP [41, p.227-229]. Letting $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{L}\boldsymbol{L}^\top = \left(\boldsymbol{Q} + \boldsymbol{a}\boldsymbol{a}^\top\right)^{-1}$ (obtained for example from a Cholesky decomposition), then point $(\boldsymbol{x}, z, w)$ satisfies constraints (13) if and only if there exists $\tau_1, \tau_2 \in \mathbb{R}_+$, $s \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}^n$ such that the constraints

$$\begin{aligned} &\tau = \tau_1 + \tau_2 \\ &\boldsymbol{L}\boldsymbol{u} + \frac{\boldsymbol{Q^{-1}a}}{\sqrt{1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2}} s = \boldsymbol{x} \\ &s\sqrt{1 + \left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2} = w \\ &\|\boldsymbol{u}\|_2^2 \le \tau_1 \\ &s^2 \le \tau_2 z \end{aligned} \tag{14}$$

are satisfied.

Projection In system (14), we can project out $\tau = \tau_1 + \tau_2$, $s = w/\sqrt{1 + \left\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\right\|_2^2}$ and $\boldsymbol{u} = \boldsymbol{L^{-1}}\left(\boldsymbol{x} - \frac{\boldsymbol{Q^{-1}a}}{1+\|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} w\right)$, which by replacing in (12) results in the formulation

$$\begin{aligned} \text{cl conv}\left(\hat{Y}_{c,\boldsymbol{Q}}\right) = \Big\{ (\boldsymbol{x}, z, t, w) \in \mathbb{R}^{n+3} \colon \exists \tau_1, \tau_2 \in \mathbb{R}_+ \text{ s.t.} \\ t \ge c^2 + 2c\left(w - \boldsymbol{a}^\top \boldsymbol{x}\right) + \tau_1 + \tau_2 \\ \left\| \boldsymbol{L^{-1}} \left(\boldsymbol{x} - \frac{\boldsymbol{Q^{-1}a}}{1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2} w \right) \right\|_2^2 \le \tau_1 \\ w^2 / \left(1 + \|\boldsymbol{Q}^{-1/2}\boldsymbol{a}\|_2^2\right) \le \tau_2 z \Big\}. \end{aligned} \tag{15}$$

In order to satisfy the first inequality constraint, we can assume that τ_1 and τ_2 are set to their lower bounds, concluding the proof. □

Remark 1. Theorem 1 reveals an interesting connection between cl conv$(\hat{Y}_{c,\boldsymbol{Q}})$ and the perspective reformulation. Indeed, letting $\boldsymbol{e_{n+1}}$ denote the $(n+1)$th standard basis vector of $\mathbb{R}^{n+1}$, one can rewrite the quadratic expression in (11) as

$$\begin{pmatrix} \boldsymbol{x}^\top & w \end{pmatrix} \boldsymbol{Q_1} \begin{pmatrix} \boldsymbol{x} \\ w \end{pmatrix} = \delta w^2 + \begin{pmatrix} \boldsymbol{x}^\top & w \end{pmatrix} \left(\boldsymbol{Q_1} - \delta \boldsymbol{e_{n+1}} \boldsymbol{e_{n+1}}^\top \right) \begin{pmatrix} \boldsymbol{x} \\ w \end{pmatrix},$$

where $\delta \ge 0$ and $\boldsymbol{Q_1} - \delta \boldsymbol{e_{n+1}} \boldsymbol{e_{n+1}}^\top \succeq 0$, and then reformulate term δw^2 as $\delta w^2/z$. From Theorem 1, we see that this reformulation is indeed ideal if δ is maximal, and the theorem provides a closed-form expression for the resulting $\boldsymbol{Q_1} - \delta \boldsymbol{e_{n+1}} \boldsymbol{e_{n+1}}^\top$ (that depends on the factorization $\boldsymbol{L}\boldsymbol{L}^\top$).

4 Application to LTS

In this section, we use the convexification results in §3 to obtain conic reformulations of (LTS+L2), or equivalently, problem (2).

4.1 The Big-M formulation

The starting point for the formulations presented is the big-M formulation

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}} & \sum_{i=1}^{m} \left(y_i + w_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 + \lambda \|\boldsymbol{T}\boldsymbol{x}\|_2^2 \\ \text{s.t. } & \mathbf{1}^\top \boldsymbol{z} \leq m - h \\ & -M\boldsymbol{z} \leq \boldsymbol{w} \leq M\boldsymbol{z} \\ & \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{z} \in \{0,1\}^m,\ \boldsymbol{w} \in \mathbb{R}^m, \end{aligned} \tag{Big-M}$$

where M is a suitably large number. Observe that while the formulation is different from the original big-M formulation (3) proposed in [56], they are equivalent in terms of strength. Indeed, variables $\boldsymbol{u}$ in (3) corresponds to terms $|y_i + w_i - \boldsymbol{a_i}^\top \boldsymbol{x}|$ in (Big-M), and the absolute values of variables $\boldsymbol{w}$ in (Big-M) can be interpreted as the slacks associated with constraints $|y_i - \boldsymbol{a_i}^\top \boldsymbol{x}| - u_i \leq M z_i$ in (3). We point out that formulation (Big-M) was the basis for the solution approach in [31] for problems with both outliers and sparsity. Indeed, the authors proposed to directly add constraints of the form $-\bar{M}\boldsymbol{\zeta} \leq \boldsymbol{x} \leq \bar{M}\boldsymbol{\zeta}$, $\mathbf{1}^\top \boldsymbol{\zeta} \leq k$ and $\boldsymbol{\zeta} \in \{0,1\}^n$ to (Big-M) – note that in [31], the regularization term $\lambda \|\boldsymbol{T}\boldsymbol{x}\|_2^2$ appeared in as constraint instead of as a penalty.

4.2 The simple conic reformulation

Observing that the objective of (Big-M) can be written as

$$\sum_{i=1}^{m} \left((y_i + w_i - \boldsymbol{a_i}^\top \boldsymbol{x})^2 + \frac{\lambda}{m} \|\boldsymbol{T}\boldsymbol{x}\|_2^2 \right),$$

we use Theorem 1 to independently reformulate each term in the sum, resulting in the formulation

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}} & \|\boldsymbol{y}\|_2^2 - 2\boldsymbol{y}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{w}) + \sum_{i=1}^{m} \left\| \boldsymbol{L_i^{-1}} \left(\boldsymbol{x} - \frac{(m/\lambda)(\boldsymbol{T}^\top \boldsymbol{T})^{-1}\boldsymbol{a}}{1 + (m/\lambda)\|(\boldsymbol{T}^\top \boldsymbol{T})^{-1/2}\boldsymbol{a}\|_2^2} \cdot w_i \right) \right\|_2^2 \\ & + \sum_{i=1}^{m} \frac{1}{1 + (m/\lambda)\|(\boldsymbol{T}^\top \boldsymbol{T})^{-1/2}\boldsymbol{a}\|_2^2} \cdot \frac{w_i^2}{z_i} \\ \text{s.t. } & \mathbf{1}^\top \boldsymbol{z} \leq m - h \\ & \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{z} \in \{0,1\}^m,\ \boldsymbol{w} \in \mathbb{R}^m, \end{aligned} \tag{conic}$$

where matrices $\boldsymbol{L_i}$ satisfy $\boldsymbol{L_i}\boldsymbol{L_i}^\top = \left((m/\lambda)\boldsymbol{T}^\top\boldsymbol{T} + \boldsymbol{a_i}\boldsymbol{a_i}^\top\right)^{-1}$. Formulation (conic) does not use the big-M constraints $-M\boldsymbol{z} \leq \boldsymbol{w} \leq M\boldsymbol{z}$, as the conic terms w_i/z_i enforce the same logical relationship. Observe that since terms x_i^2/z_i can be reformulated as SOCP-constraints [2], and every other term is either convex quadratic or linear, formulation (conic) can be easily used with mixed-integer SOCP solvers.

4.3 The stronger conic reformulation

The observation motivating the stronger conic reformulation is that, given any collection of matrices $\{\boldsymbol{Q_i}\}_{i=1}^m$ such that $\boldsymbol{Q_i} \succ 0$ and $\sum_{i=1}^m \boldsymbol{Q_i} = \lambda\boldsymbol{T}^\top\boldsymbol{T}$, we can rewrite the objective of (Big-M) as

$$\sum_{i=1}^m \left((y_i + w_i - \boldsymbol{a_i}^\top\boldsymbol{x})^2 + \boldsymbol{x}^\top\boldsymbol{Q_i}\boldsymbol{x}\right)$$

and then apply Theorem 1. The simple conic reformulation is a special case of such a convexification, with $\boldsymbol{Q_i} = (\lambda/m)\boldsymbol{T}^\top\boldsymbol{T}$ for all $i \in [m]$, but other choices of collection $\{\boldsymbol{Q_i}\}_{i=1}^m$ may result in stronger formulations. We now discuss how to find a collection $\{\boldsymbol{Q_i}\}_{i=1}^m$ resulting in better relaxations.

We use the intuition provided in Remark 1 and similar ideas to [16, 55] to derive the formulation. Observe that given any collection $\{\boldsymbol{Q_i}\}_{i=1}^m$, the relaxation is of the form

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}} \ & \|\boldsymbol{y}\|_2^2 - 2\boldsymbol{y}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{w}) + \begin{pmatrix}\boldsymbol{x}^\top & \boldsymbol{w}^\top\end{pmatrix}\boldsymbol{\Sigma}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{w}\end{pmatrix} + \sum_{i=1}^m d_i\frac{w_i^2}{z_i} \\
\text{s.t.} \ & \sum_{i=1}^m z_i \leq m - h \\
& \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in [0,1]^m, \boldsymbol{w} \in \mathbb{R}^m,
\end{aligned}$$

where $\boldsymbol{\Sigma} \succeq 0$ and $\boldsymbol{d} \geq \boldsymbol{0}$. Moreover, we find that

$$\boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{T}^\top\boldsymbol{T} & -\boldsymbol{A}^\top \\ -\boldsymbol{A} & \boldsymbol{I} - \mathrm{Diag}(\boldsymbol{d})\end{pmatrix}.$$

Thus, the continuous relaxation of the stronger conic reformulation is given by

$$\begin{aligned}
\max_{\boldsymbol{d},\boldsymbol{\Sigma}} \ & \min_{\substack{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{w})\in\mathbb{R}^{n+2m} \\ \|\boldsymbol{z}\|_1 \leq m-h \\ \boldsymbol{0}\leq\boldsymbol{z}\leq\boldsymbol{1}}} \|\boldsymbol{y}\|_2^2 - 2\boldsymbol{y}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{w}) + \begin{pmatrix}\boldsymbol{x}^\top & \boldsymbol{w}^\top\end{pmatrix}\boldsymbol{\Sigma}\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{w}\end{pmatrix} + \sum_{i=1}^m d_i\frac{w_i^2}{z_i} \\
\text{s.t.} \ & \boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{T}^\top\boldsymbol{T} & -\boldsymbol{A}^\top \\ -\boldsymbol{A} & \boldsymbol{I} - \mathrm{Diag}(\boldsymbol{d})\end{pmatrix} \succeq 0 \\
& \boldsymbol{d} \in \mathbb{R}^m_+, \ \boldsymbol{\Sigma} \in \mathbb{R}^{(n+m)\times(n+m)}.
\end{aligned} \tag{conic+}$$

Observe that in formulation (conic+), constraints $\boldsymbol{z} \in \{0,1\}$ were relaxed to bound constraints, hence it is a relaxation of (2). The MIO version corresponds to fixing $\boldsymbol{d}$ and $\boldsymbol{\Sigma}$ to the optimal values of (conic+), and adding back constraints $\boldsymbol{z} \in \{0,1\}^m$.

We now discuss how to compute an optimal solution $\boldsymbol{d}^*$ of (conic+) – note that $\boldsymbol{\Sigma}^*$ is immediately implied from the value of $\boldsymbol{d}^*$. Given any fixed $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{w}}) \in \mathbb{R}^n \times [0,1]^m \times \mathbb{R}^m$, the choice of $\boldsymbol{d}$ that results in the best relaxation for that particular point (i.e., resulting in the largest objective value for that particular point with $\boldsymbol{z}$ fractional) is an optimal solution of the semidefinite optimization problem (where we remove terms that do not depend on $\boldsymbol{d}$)

$$\max_{\boldsymbol{d} \in \mathbb{R}^m_+} \sum_{i=1}^m \bar{w}_i^2 \left(\frac{1}{\bar{z}_i} - 1\right) d_i \tag{17a}$$

$$\text{s.t. } \begin{pmatrix} \boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{T}^\top \boldsymbol{T} & -\boldsymbol{A}^\top \\ -\boldsymbol{A} & \boldsymbol{I} - \text{Diag}(\boldsymbol{d}) \end{pmatrix} \succeq 0. \tag{17b}$$

While convex and polynomial-time solvable, problem (17) can be difficult to solve, mainly due to the presence of the large-dimensional conic constraint (17b), on order $(n+m)$ matrices. Fortunately, as Proposition 3 below shows, problem (17) can be reformulated using a lower dimensional conic constraint, on order n matrices.

Proposition 3. *If $\boldsymbol{A}$ does not contain a row of 0s and $\boldsymbol{u}^*$ is optimal for the optimization problem*

$$\begin{aligned} \min_{\boldsymbol{u} \in \mathbb{R}^m} & \sum_{i=1}^m \bar{w}_i^2 \left(\frac{1}{\bar{z}_i} - 1\right) \frac{1}{u_i} \\ & \textit{s.t. } \boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{T}^\top \boldsymbol{T} - \boldsymbol{A}^\top \textit{Diag}(\boldsymbol{u}) \boldsymbol{A} \succeq 0,\ \boldsymbol{u} \geq \boldsymbol{1}, \end{aligned} \tag{18}$$

then $\boldsymbol{d}^* \in \mathbb{R}^m_+$ such that $d_i^* = 1 - \frac{1}{u_i^*}$ is optimal for (17).

Proof. From the generalized Schur complement [3], we find that constraint (17b) is equivalent to

$$\boldsymbol{I} - \text{Diag}(\boldsymbol{d}) \succeq 0, \tag{19a}$$

$$(\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))\,(\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))^\dagger\, \boldsymbol{A} = \boldsymbol{A}, \text{ and} \tag{19b}$$

$$\boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{T}^\top \boldsymbol{T} - \boldsymbol{A}^\top\, (\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))^\dagger\, \boldsymbol{A} \succeq 0. \tag{19c}$$

Constraint (19a) is equivalent to $\boldsymbol{d} \leq \boldsymbol{1}$. Constraint (19b) is automatically satisfied if $\boldsymbol{d} < \boldsymbol{1}$, since in that case matrix $(\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))^\dagger = (\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))^{-1}$. In general, however, $\Omega = (\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))\,(\boldsymbol{I} - \text{Diag}(\boldsymbol{d}))^\dagger$ is the diagonal matrix such that $\Omega_{ii} = \mathbb{1}_{\{d_i < 1\}}$. Therefore, if $d_i = 1$, then the i-th row of matrix ΩA is a row of 0s, and constraint (19b) cannot be satisfied in that case unless the i-th row of $\boldsymbol{A}$ is also $\boldsymbol{0}$.

Finally, perform a change of variables $u_i = \frac{1}{1-d_i}$, well defined since $d_i < 1$ holds. From constraints $\boldsymbol{d} \geq 0$ we find $\boldsymbol{u} \geq \mathbf{1}$. Problem (17) reduces to

$$\begin{aligned} \max_{\boldsymbol{u} \in \mathbb{R}^m} \; & \sum_{i=1}^{m} \left(\bar{w}_i^2 \left(\frac{1}{\bar{z}_i} - 1 \right) - \bar{w}_i^2 \left(\frac{1}{\bar{z}_i} - 1 \right) \frac{1}{u_i} \right) \\ \text{s.t.} \; & \boldsymbol{A}^\top \boldsymbol{A} + \lambda \boldsymbol{T}^\top \boldsymbol{T} - \boldsymbol{A}^\top \mathrm{Diag}(\boldsymbol{u}) \boldsymbol{A} \succeq 0, \; \boldsymbol{u} \geq \mathbf{1}. \end{aligned}$$

The result then follows by removing terms in the objective not involving $\boldsymbol{u}$. □

Remark 2. The assumption on $\boldsymbol{A}$ is almost without loss of generality, since it is invariably satisfied in practice. In the formulation in Proposition 3, the nonlinear objective terms can reformulated with the introduction of additional variables $\boldsymbol{s} \in \mathbb{R}^m_+$ and rotated cone constraints $1 \leq s_i u_i$. The formulation contains a similar number of variables as (17), but if $n \ll m$ the nonlinear conic constraints are much simpler, and as a consequence the resulting formulation is substantially faster (and less memory intensive as well).

We propose a simple primal-dual method to solve (conic+), summarized in Algorithm 1. The algorithm iterates between solving the inner minimization of (conic+) to optimality (for fixed $\boldsymbol{d}$ and $\boldsymbol{\Sigma}$), and moving towards the optimal of the outer maximization (for fixed $\boldsymbol{x}$, $\boldsymbol{z}$ and $\boldsymbol{w}$). Each minimization step requires solving an SOCP-representable problem, while each maximization step requires solving an SDP as outlined in Proposition 3. The final MIO can be solved with off-the-shelf mixed-integer SOCP solvers.

Algorithm 1 (conic+) algorithm.

$k \leftarrow 0$ ▷ Iteration number
$d_i^0 \leftarrow \frac{1}{1+(m/\lambda)\left\|(\boldsymbol{T}^\top \boldsymbol{T})^{-1/2} \boldsymbol{a_i}\right\|_2^2}$ for $i = 1, \ldots, m$ ▷ $\boldsymbol{d^0}$ from formulation (conic)
repeat
  $k \leftarrow k + 1$
  $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{w}}) \leftarrow$ `Solve` inner minimization of (conic+) with $\boldsymbol{d} = \boldsymbol{d^{k-1}}$ fixed
  $\boldsymbol{d^*} \leftarrow$ `Solve` (17) ▷ Use Proposition 3
  $\boldsymbol{d^k} \leftarrow \boldsymbol{d^{k-1}} + \frac{1}{k}\left(\boldsymbol{d^*} - \boldsymbol{d^{k-1}}\right)$
until Termination criterion is met
$(\boldsymbol{x^*}, \boldsymbol{z^*}, \boldsymbol{w^*}) \leftarrow$ `Solve` (conic+) with constraints $\boldsymbol{z} \in \{0,1\}^m$ and $\boldsymbol{d} = \boldsymbol{d^k}$ fixed.
return $(\boldsymbol{x^*}, \boldsymbol{z^*}, \boldsymbol{w^*})$

4.4 Improving relaxations with reliable data

In some situations, a decision-maker may have access to data that has been carefully vetted, and is known to be reliable. Obviously, in such situations, such data should not be discarded. Moreover, as we now discuss, it is possible to leverage such data to further improve the relaxations.

Suppose that the first m_0 datapoints are known to not contain outliers. Then, (Big-M) simplifies to

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}} & \sum_{i=1}^{m_0} \left(y_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 + \sum_{i=m_0+1}^{m} \left(y_i + w_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 + \lambda \|\boldsymbol{T}\boldsymbol{x}\|_2^2 \\ \text{s.t. } & \mathbf{1}^\top \boldsymbol{z} \leq m - h \\ & -M\boldsymbol{z} \leq \boldsymbol{w} \leq M\boldsymbol{z} \\ & \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{z} \in \{0,1\}^{m-m_0},\ \boldsymbol{w} \in \mathbb{R}^{m-m_0}.\end{aligned}$$

Expanding the error terms of first m_0 points, we may rewrite the objective as

$$\sum_{i=1}^{m_0} \left(y_i^2 - 2y_i \boldsymbol{a_i}^\top \boldsymbol{x}\right) + \sum_{i=m_0+1}^{m} \left(y_i + w_i - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 + \boldsymbol{x}^\top \left(\lambda \boldsymbol{T}^\top \boldsymbol{T} + \sum_{i=1}^{m_0} \boldsymbol{a_i} \boldsymbol{a_i}^\top\right) \boldsymbol{x}.$$

In other words, matrix $\lambda \boldsymbol{T}^\top \boldsymbol{T} + \sum_{i=1}^{m_0} \boldsymbol{a_i} \boldsymbol{a_i}^\top$ can be treated as the "regularization" matrix and used throughout the conic formulations instead of $\lambda \boldsymbol{T}^\top \boldsymbol{T}$, resulting in stronger formulations.

Remark 3. Note that even if no reliable data is available, the ideas here can still be used to improve algorithms. For example, in a branch-and-bound search, at any given node some subset of variables $\boldsymbol{z}$ may have been fixed to $\mathbf{0}$. Thus, we may use the ideas in this subsection to improve the relaxations for the subtree emanating from that node. Doing so, however, would require a large degree of control of the branch-and-bound algorithm, which is not possible for several off-the-shelf branch-and-bound solvers.

4.5 Intercept

The presence of the strictly convex term $\|\boldsymbol{T}\boldsymbol{x}\|_2^2$ is critical for the design of strong convex relaxations (Proposition 1). However, the presence of an intercept variable might hamper the exploitation of the regularization term. Indeed, while the intercept is often subsumed into matrix $\boldsymbol{A}$ (as a column of 1s), the regularization term rarely involves the intercept variable. Indeed, writing (Big-M) while making the intercept variable x_0 explicit results in

$$\begin{aligned}\min_{x_0,\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}} & \sum_{i=1}^{m} \left(y_i + w_i - x_0 - \boldsymbol{a_i}^\top \boldsymbol{x}\right)^2 + \lambda \|\boldsymbol{T}\boldsymbol{x}\|_2^2 \\ \text{s.t. } & \mathbf{1}^\top \boldsymbol{z} \leq m - h \\ & -M\boldsymbol{z} \leq \boldsymbol{w} \leq M\boldsymbol{z} \\ & (x_0, \boldsymbol{x}) \in \mathbb{R}^{n+1},\ \boldsymbol{z} \in \{0,1\}^m,\ \boldsymbol{w} \in \mathbb{R}^m,\end{aligned}$$

where the intercept x_0 does not appear in term $\|\boldsymbol{T}\boldsymbol{x}\|_2^2$, and thus this term is not strictly convex. Observe that the quadratic objective function is rank-deficient, as the rank of the quadratic function is at most $m+n$, while there are $n+m+1$

variables $(x_0, \boldsymbol{x}, \boldsymbol{w})$. As a consequence the conic formulations may be ineffective, e.g., feasible solutions of optimization (17) may require $d_i = 0$ for at least some index i.

We propose three workarounds to resolve the difficulties posed by the intercept. The first one, which is the ideal solution, is to use reliable data as discussed in §4.4: a single datapoint known to be reliable will allow for the application of the conic formulations. The second approach, which is common in practice, is to standardize $\boldsymbol{y}$ so that it has 0-mean, and fix $x_0 = 0$. In other words, fix the intercept to be the mean of the response variable. The third approach is to artificially create a strictly quadratic term involving the intercept as a regularization. In particular, given a baseline value $\bar{c}_0$ for the intercept (e.g., obtained from a heuristic solution), add a regularization term $\lambda(x_0 - \bar{c}_0)^2$ to the objective, penalizing departure of x_0 from this baseline. Naturally, the addition of this regularization may prevent the resulting formulation from finding optimal solutions of (2). Nonetheless, in our computations, we found that the solutions obtained from the conic formulations are still high quality even if the baseline value $\bar{c}_0$ is poorly chosen.

5 Computations

In this section, we discuss computations with synthetic and real data. First, in §5.1, we discuss the different methods compared. Then in §5.2 we provide a high level summary of our computational results, in §5.3 we discuss experiments with synthetic data, used to validate the statistical merits of the approach, and in §5.4 we provide experiments with real datasets.

5.1 Methods tested

We compare the three MIO formulations presented in §4 for regression problems with outliers.

`big-M` The big M method, as discussed in §4.1.

`conic` The simple conic reformulation, as discussed in §4.2.

`conic+` The stronger conic reformulation, as discussed in Algorithm 1 in §4.3.

In addition, we also compare the following commonly used methods.

`ls+l2` Simply solving problem (`LS+L2`), as described in §1, without accounting for outliers.

`lad` Solving the least absolute deviation problem (`LAD`).

`alt-opt` Heuristic that alternates between optimizing regression coefficients $\boldsymbol{x}$ given a fixed set of $m - h$ discarded observations (by fitting an `ls+l2` regression), and optimizes which $m - h$ observations to discard (encoded by $\boldsymbol{z}$) given fixed regression coefficients (a process called a C-step in [46]). We set the initial regression coefficients to be those obtained from (`LS+L2`).

We point out that we also attempted to implement the MIO formulation of [11] for least quantile regression. However that formulation, which includes three different sets of big-M constraints, resulted in numerical issues in most of the instances (with the solver producing "optimal" solutions that are not feasible for the MIO, and are extremely poor estimators). In any case, as mentioned in §1, the authors in [11] comment that the solutions produced by the MIO formulation are much worse than heuristic solutions (unless warm-started with such solutions, in which case the quality of solutions produced by MIO matches the heuristics).

In all cases we use the ridge regularization $\boldsymbol{T} = \boldsymbol{I}$. We use Gurobi solver 9.5 to solve all (mixed-integer) linear, quadratic or second order cone optimization problems, and solver Mosek 10.0 to solve SDPs. All computations are done on a laptop with a 12th Gen Intel Core i7-1280 CPU and 32 GB RAM. We set a time limit of 10 minutes for all methods (for `conic+`, this time includes both solving SDPs and a MIO), and use the default configuration of the solvers in all cases. All instances are standardized, that is, the model matrix $\boldsymbol{A}$ is translated and scaled so that $\sum_{i=1}^m A_{ij} = 0$ and $\sum_{i=1}^m A_{ij}^2 = 1$ for all $j \in [n]$; similarly, $\sum_{i=1}^m y_i = 0$ and $\sum_{i=1}^m y_i^2 = 1$.

5.1.1 Implementation details and numerical considerations

We now discuss how we select and tune parameters for the different methods, as well as discuss potential issues if the parameters are poorly chosen.

`big-M` Formulation (Big-M) depends on the parameter M. If a small value is chosen, then the formulation might remove optimal solutions. If the value chosen is too large, then numerical issues can be encountered: for example, the solver might set $z_j = 10^{-5}$ for some j (which is interpreted as 0 due to numerical precision of solvers) but set w_j to a large value, while satisfying constraint $|w_j| \leq M z_j$. In our experiments we set $M = 1,000$, and we did not observe any numerical issue in our experiments. This parameter was not tuned.

`conic` Formulation (conic) does not involve any parameter, however it requires to have $\lambda > 0$ and may result in incorrect behavior if $\lambda \to 0$. Moreover, based on past experience by the authors, mixed-integer SOCP formulations may result in poor performance or numerical difficulties in large problems. Our experiments satisfy $\lambda \geq 0.01$ and we did not observe any numerical issues.

To handle the intercept (recall the discussion in §4.5), we tested both fixing $x_0 = 0$, or using the intercept $\bar{x}_0$ produced by `ls+l2` as a proxy, and adding the regularization term $\lambda(\bar{x}_0 - x_0)^2$ to the objective (where λ is the same coefficient as the one appearing in term $\lambda\|\boldsymbol{x}\|_2^2$). We did not observe major differences between the two approaches, in terms of quality or solution times. In our experiments with real data we set $x_0 = 0$, and in our experiments with synthetic data we use the value of `ls+l2` as a proxy (not that the synthetic experiment includes the instances where `ls+l2` performs worse).

`conic+` As the most sophisticated formulation, there are several implementation details associated with method `conic+`. First, note that if an optimal solution of problem (18) satisfies $u_i^* = 1$ for some index $i \in [m]$, then $d_i^* = 0$ and formulation (conic+) does not include term w_i^2/z_i (thus the MIO formulation is not exact, but rather a relaxation). Thus, in our computations, we set a constraint $u_i \geq 1.001$. We did not tune this lower bound, although we noted that simply setting $u_i \geq 1$ does indeed result in incorrect results.

We now discuss the termination criterion of Algorithm 1. Note that at each iteration, at line 5 of the algorithm, a lower bound on the optimal objective value of (2) is computed. Moreover, given the solution to the relaxation $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{w}})$, we can compute an upper bound on the optimal objective value using a rounding heuristic, by setting $z_j = 1$ for indexes corresponding to the $m - h$ largest values of $\boldsymbol{z}$, and then solving (2) with $\boldsymbol{z}$ fixed. Neither the sequence of lower and upper bounds produced by the algorithm is guaranteed to be monotonic, so we track the best lower (LB) and upper bound (UB) found throughout all previous iterations, and compute an optimality gap at any given iteration as $\text{gap} = (UB - LB)/UB$. Finally, we stop the algorithm after 20 iterations (not necessarily consecutive) in which the gap improvement from one iteration to the next is less than 10^{-6}. The parameters $(20, 10^{-6})$ were tuned minimally, based on one synthetic instance and one real instance with the alcohol dataset (and on these datasets we did not observe major differences for different choices of parameters).

In terms of numerical difficulties, in addition to those already mentioned for the `conic` formulation, method `conic+` requires solving several SDPs with low-dimensional cones. Certainly, SDPs are inherently more difficult than quadratic or conic quadratic problems, more sensitive to the input data and more prone to numerical instabilities. For example, we observed that if the raw data $(\boldsymbol{A}, \boldsymbol{y})$ is used without standardization, the SDP solver encounters numerical difficulties in several of the instances. In our experiments, with standardized data, we did encounter numerical issues in a single instance (out of over 400). The intercept is handled similarly to `conic`.

5.1.2 A note on additional improvements

We point out that all methods presented here can be further improved. For example, one might use the solution of any of the heuristic methods as warm start for the MIO, and after run heuristic `alt-opt` starting from the solution produced by the MIO (assuming the time limit was reached, in which case the solution of the MIO might not be optimal), which will produce a solution that is at least as good as the solutions obtained from either solving the MIO or using heuristic `alt-opt` independently. We do suggest practitioners to use such improvements in practice. However, we point out that our objective in the paper is not to propose an algorithm that is "best" for the LTS problem, but rather to evaluate the strength of the conic formulations presented (which, as pointed out in §1, might be used as building blocks for other optimization problems). By not including additional improvements, we ensure that the differences in

computational performance between the MIO methods is entirely due to the formulations used.

5.2 Summary of results

We first provide a summary of the results in our computations.

• In computations with both synthetic and real datasets, formulations `conic` and `conic+` are orders-of-magnitude faster than `big-M`.

• In computations with both synthetic and real datasets, heuristic `alt-opt` is very fast and delivers optimal solutions in a good proportion of instances, but produces extremely poor solutions in the remaining instances (often worse than solutions obtained by not handling outliers at all). The exact formulation `conic+` consistently produces high-quality solutions (even in instances that are not solved to optimality).

• In our computations, formulation `conic+` solves all synthetic instances in a few seconds –including instances with $(n, m) \in \{20, 500\}$– but fails to solve within the time limit real instances with $(n, m) \in \{10, 60\}$. Therefore, we advocate for departing from the common practice in the statistical and machine learning literature to use synthetic instances to evaluate the scalability of exact methods for (`LTS+L2`) or related problems.

5.3 Experiments with synthetic data

We now discuss experiments with synthetic data. First we discuss the instance generation process in §5.3.1 and the relevant metrics in §5.3.2, then we present computational performance in §5.3.3 and statistical results in §5.3.4. We point out that the focus in this section is the statistical performance of the different methods.

5.3.1 Instance generation

Our instance generation process follows closely the generation process in [11], which in turn was inspired by [46]. Given parameters n and m, each entry of the matrix $\boldsymbol{A}$ is generated iid as $A_{ij} \sim \mathcal{N}(0, 100)$. Moreover, we generate a "ground-truth" vector $\boldsymbol{x}^* = \mathbf{1}$, and responses as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}$, where each entry of $\boldsymbol{\epsilon}$ is generated iid as $\epsilon_i \sim \mathcal{N}(0, 10)$. Given a proportion τ of outliers, we randomly choose $\lfloor \tau m \rfloor$ of the observations as outliers and modify the associated responses as $y_i \leftarrow y_i + 1,000$. Finally, data is standardized.

In all our experiments, we set the budget of outliers $m - h$ to be equal to the proportion of outliers $\lfloor \tau m \rfloor$ (as done in [11, 46]).

5.3.2 Metrics

In this section we compare the statistical benefits of the different methods. To assess the quality of a given method, we compare two metrics. Given an estimate

$\hat{\boldsymbol{x}}$ for a given method, the relative risk defined as

$$\texttt{risk} = \frac{\|\boldsymbol{x}^* - \hat{\boldsymbol{x}}\|_2^2}{\|\boldsymbol{x}^*\|_2^2},$$

where $\boldsymbol{x}^* = \mathbf{1}$ is the ground truth, measures the error of the estimated coefficients. A perfect prediction results in $\texttt{risk} = 0$, the naive predition $\hat{\boldsymbol{x}} = \mathbf{0}$ results in $\texttt{risk} = 1$, and method with low breakdown point (e.g., `ls+l2`) may result in arbitrarily large values of `risk`. Given a set of candidate outliers encoded by an indicator vector $\hat{\boldsymbol{z}}$, the recall defined as

$$\texttt{recall} = \frac{|\{i \in [m] : \hat{z}_i = 1 \text{ and } i \text{ is an outlier}\}|}{\lfloor \tau m \rfloor}$$

measures the proportion of outliers that are correctly identified. Note that `recall` is not defined for `lad` and `ls+l2`, since those methods do not explicitly identify outliers.

5.3.3 Computational performance

We test small instances for all combinations of parameters $n \in \{2, 20\}$, $m \in \{100, 500\}$, $\lambda \in \{0.01, 0.1, 0.2, 0.3\}$ and $\tau \in \{0.1, 0.2, 0.4\}$. For each combination of parameters, we generate five instances. Heuristics such as `lad` and `alt-opt` are solved in a fraction of a second. The results for MIO formulations are summarized in Figure 2. We observe that method `big-M` can solve 50% of the instances in under 10 minutes, a performance comparable with the one reported in [31] (although the data generation process is different). Methods `conic` and `conic+` are much faster. In particular, `conic+` can solve all instances in less than 11 seconds, resulting in at least a two-orders-of-magnitude speedup over `big-M`.

As we observe in our computations with real datasets (see §5.4), the results here are not representative of the actual performance of the methods in practice. Therefore, we do not provide detailed computational results in this section. We simply comment on the effect of parameters τ and λ: instances with small number of outliers $\lfloor \tau m \rfloor$ are much easier to solve (most of instances solved to optimality by `big-M` correspond to small values of τ), and formulation `conic` benefits from larger values of regularization λ as well. Finally, the continuous relaxation of `conic+` is strong regardless of the combination of parameters, and most of the instances are solved at the root node.

5.3.4 Statistical results

We now present the statistical performance for different methods for parameters $(n, m) \in \{(2, 100), (20, 100), (20, 500)\}$. We omit results for methods `big-M` and `conic`, since `conic+` delivers similar solutions much faster. In Table 1, we compare the performance of `lad`, and methods `conic+`, `alt-opt` and `ls+l2` with parameter $\lambda = 0.01$. The results for $m = 100$ are also summarized in Figure 3. Table 2 shows the effect of varying the parameter λ for the relative risk of estimators `conic+`, `alt-opt` and `ls+l2`.

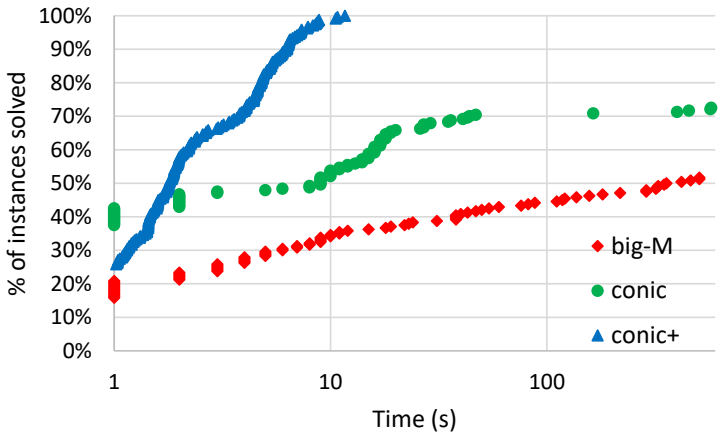


Figure 2: Percentage of synthetic instances solved as a function of time (in log scale). Method `big-M` can solve 52% of instances in 600 seconds, while formulations `conic` and `conic+` require 9 seconds and 2 seconds to solve the same quantity of instances, respectively, thus resulting in 66x and 300x speed-ups in those instances.

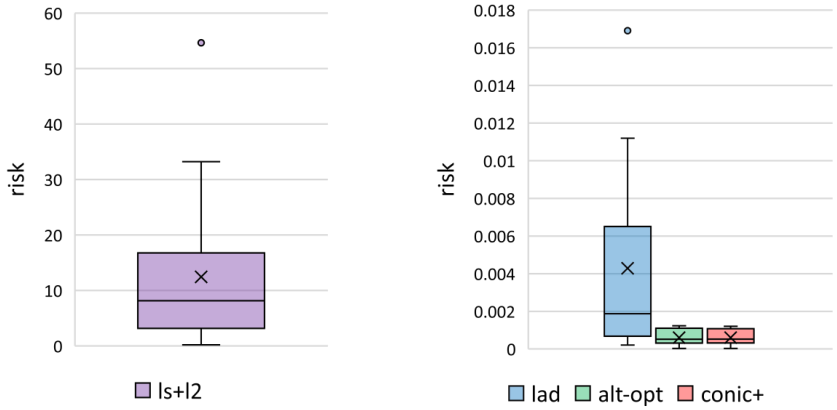
Table 1: Statistical performance for different regression methods. We use regularization parameter $\lambda = 0.01$ for `conic+`, `alt-opt` and `ls+l2`, while `lad` have no regularization. Each row represents the average over five instances generated with identical parameters. We show in **bold** metrics that are the best for that particular class of instances.

n	m	τ	conic+		alt-opt		lad		ls+l2	
			risk	recall	risk	recall	risk	recall	risk	recall
2	100	0.1	**0.001**	**1.00**	**0.001**	**1.00**	**0.001**	-	6.387	-
2	100	0.2	**0.001**	**1.00**	**0.001**	**1.00**	0.002	-	7.443	-
2	100	0.4	**0.001**	**1.00**	**0.001**	**1.00**	0.009	-	23.528	-
20	100	0.1	**0.001**	**1.00**	**0.001**	**1.00**	0.003	-	12.262	-
20	100	0.2	**0.002**	**1.00**	**0.002**	**1.00**	0.005	-	19.388	-
20	100	0.4	**0.004**	**1.00**	109.979	0.65	83.271	-	26.580	-
20	500	0.1	**0.000**	**1.00**	**0.000**	**1.00**	**0.000**	-	1.631	-
20	500	0.2	**0.000**	**1.00**	**0.000**	**1.00**	0.001	-	3.234	-
20	500	0.2	**0.001**	**1.00**	**0.001**	**1.00**	0.003	-	4.695	-

We note that estimator `conic+` results in the best risk and recall in all instances considered, and is the only estimator which does not break down in instances with $n = 20$, $m = 100$ and $\tau = 0.4$ (i.e., instances with the smallest signal-to-noise ratio and larger number of outliers among those considered). We observe that formulation `ls+l2` in general produces poor solutions, as expected. Indeed, with a breakdown point of 0, the presence of a single outlier could result

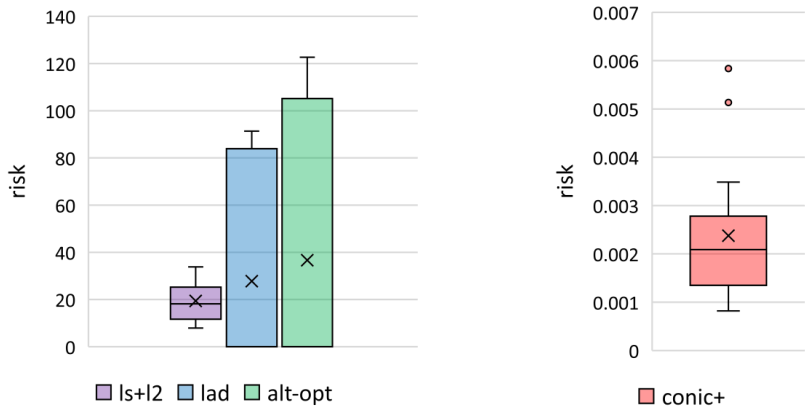
Table 2: Relative risk for `conic+`, `alt-opt` and `ls+l2`. Each row represents the average over five instances generated with identical parameters.

n	m	τ	$\lambda = 0.01$			$\lambda = 0.1$			$\lambda = 0.2$			$\lambda = 0.3$		
			`conic+`	`alt-opt`	`ls+l2`	`conic+`	`alt-opt`	`ls+l2`	`conic+`	`alt-opt`	`ls+l2`	`conic+`	`alt-opt`	`ls+l2`
2	100	0.1	0.001	0.001	6.387	0.009	0.009	5.316	0.030	0.030	4.420	0.058	0.058	3.739
2	100	0.2	0.001	0.001	7.443	0.011	0.011	6.332	0.036	0.036	5.382	0.067	0.067	4.645
2	100	0.4	0.001	0.001	23.582	0.015	0.015	19.901	0.051	0.051	16.800	0.096	0.096	14.393
20	100	0.1	0.001	0.001	12.262	0.017	0.017	9.165	0.050	0.050	7.904	0.087	0.096	5.747
20	100	0.2	0.002	0.001	19.388	0.023	0.023	14.848	0.063	0.063	11.609	0.107	0.107	9.410
20	100	0.4	0.004	109.979	26.580	0.070	23.527	20.398	0.115	0.115	15.965	0.178	0.178	12.944
20	500	0.1	0.000	0.000	1.631	0.011	0.011	1.360	0.035	0.035	1.147	0.067	0.067	0.997
20	500	0.2	0.000	0.000	3.234	0.013	0.013	2.676	0.043	0.043	2.228	0.079	0.079	1.900
20	500	0.4	0.001	0.001	4.695	0.022	0.022	3.927	0.067	0.067	3.288	0.117	0.117	2.807



(a) $n = 2$, breakdown

(b) $n = 2$, robust



(c) $n = 20$, breakdown

(d) $n = 20$, robust

Figure 3: Distributions of the relative risk of different methods for instances with $m = 100$, $\lambda = 0.01$, $\tau \in \{0.1, 0.2, 0.4\}$. The top row shows instances with $n = 2$, and the bottom row shows instances with $n = 20$. The left column depicts risk for estimators that broke down, resulting in `risk` > 1, and the right column depicts risk for estimators that produced high quality solutions (note the difference in the scale of vertical axis).

in arbitrarily poor solutions, and the instances used contain several outliers. Interestingly, while formulation `lad` also has a breakdown point of 0, it produces good solutions in most of the instances considered (although even in those instances, the relative risk can be five times more than the risk of other robust approaches). However, in instances with $n = 20$, $m = 100$ and $\tau = 0.4$, the esti-

mator breaks down and results in extremely poor solutions, worse in fact than those produced by estimator `ls+l2` which ignores outliers. Heuristic `alt-opt` matches the performance of `conic+` in instances where $(n, m, \tau) \neq (20, 100, 0.4)$ –in fact, the heuristic finds optimal solutions in all these instances–, but fails dramatically in the setting $(n, m, \tau) = (20, 100, 0.4)$: the solutions produced are poor local minima of (`LTS+L2`), and the statistical properties are in fact worse than `ls+l2` and `lad`.

We see from Table 2 that as the regularization parameter λ increases, the performance of `ls+l2` improves (showcasing how the ℓ_2 regularization induces robustness) but is still substantially worse than `conic+`. On the other hand, we see that an increase of the regularization parameter results in worse performance for `conic+`, but the risk remains low for all combinations of regularization tested. Finally we observe that in instances with $(n, m, \tau) = (20, 100, 0.4)$, an increase of regularization results in much better performance for heuristic `alt-opt`, and the estimator does not break down if $\lambda \geq 0.2$. However, the resulting risk is larger than the risk of solutions produced by `conic+` with smaller values of regularization parameter.

5.4 Experiments with real data

We now discuss experiments with real data. First we present the instances used in §5.4.1 and the metrics tested in §5.4.2, and then discuss computational times in §5.4.3 and solution quality in §5.4.4.

5.4.1 Instances

To test the methods we use instances included in the software package "robust base" [44]. Specifically, we select the instances that: *(i)* are regression instances (as opposed to classification), *(ii)* do not have missing data, and *(iii)* are not time series data. The resulting 17 datasets are summarized in Table 3. For each instance, we vary the parameter $\lambda \in \{0.05, 0.1, 0.2\}$ and the proportion of allowed outliers $m - h \in \{\lfloor 0.1m \rfloor, \lfloor 0.2m \rfloor, \lfloor 0.3m \rfloor, \lfloor 0.4m \rfloor\}$, thus creating 12 different instances for each particular dataset. Note that we separate the instances in nine "easy" datasets (all satisfying $m \leq 40$) and eight "hard" datasets (with $m > 40$). This distinction is based on the performance of the `big-M` formulation: the average time to solve all instances for an easy dataset is less than 15 seconds, whereas for the hard datasets there is at least one instance that could not be solved to optimality within the time limit of 10 minutes.

5.4.2 Metrics

On real data, there is no "ground truth" concerning which points are outliers or the actual values of the regression coefficients. Thus, we limit our comparisons to the performance of MIO algorithms (as measured by time, nodes and optimality gap) and the quality of the solutions obtained in terms of the objective value of (2) of `conic+` and `alt-opt`.

Table 3: Real datasets used. Problems in "easy" datasets can be solved to optimality by `big-M` in seconds, whereas time limits are reached for `big-M` in hard datasets.

	name	n	m
Easy	pension	1	18
	phosphor	2	18
	salinity	3	18
	toxicity	9	18
	pilot	1	20
	wood	5	20
	steamUse	4	38
	bushfire	4	38
	starsCYG	1	41
Hard	alcohol	6	44
	education	4	50
	epilepsy	9	59
	pulpfiber	7	62
	wagner	6	63
	milk	7	86
	foodstamp	3	150
	radarimage	4	1,573

5.4.3 Computational performance of MIO

Similarly to results with synthetic instances, heuristics such as `alt-opt` run in a fraction of a second in all cases. In computations with easy datasets, `big-M` solves the instances in three seconds on average, and under 45 seconds in all cases. Formulation `conic` also requires three seconds on average as well (and 87 seconds in the worst-case), while formulation `conic+` requires only one second (and under 10 seconds in the worst case). Note that easy datasets have $m \leq 41$, and full enumeration may be possible in most of the instances. In the interest of shortness, we do not present detailed results on the computations with easy datasets, and focus in this section in the more interesting computations of `MIO` methods with hard datasets.

Figure 4 presents aggregated results for instances with hard datasets. In particular, it shows the percentages of instances solved by methods `big-M`, `conic` and `conic+` within any given time limit. Observe that the performance of all methods in instances with real datasets is worse than the one reported in synthetic instances, despite real datasets being in some cases smaller by an order-of-magnitude. This discrepancy of performance serves as compelling evidence that synthetic instances should not be used to evaluate the "scalability" of MIO methods for `LTS` or related problems in regression. Indeed, as is well-known in the MIO literature, size of an instance is often a poor proxy of its difficulty.

We see that the `big-M` formulation struggles, solving only 22% of the instances within the time limit of 600 seconds. Formulation `conic+` is better across the board, requiring only 16 seconds to solve 22% of the instances, and managing to solve 35% of the instances overall. Formulation `conic+` is worse than `conic` in the simpler instances, but much better in the more difficult ones, managing to solve over 45% of the instances. Indeed, in instances that can be solved easily by other methods, the additional cost of solving SDPs hurts the performance, but the stronger relaxation pays off in difficult instances.

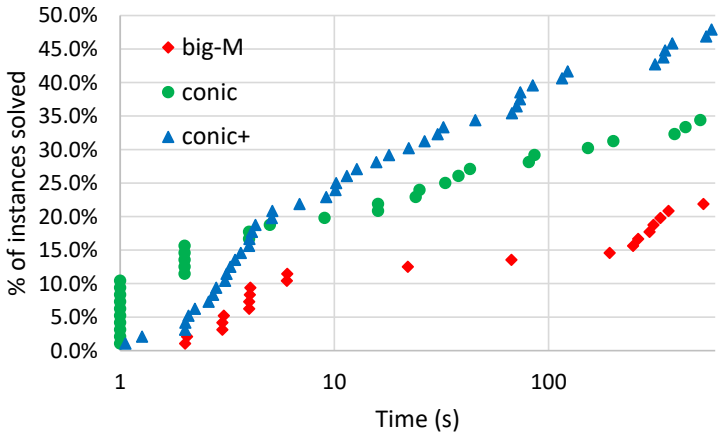


Figure 4: Percentage of instances with "hard" real datasets solved as a function of time (in log scale). Method `big-M` can solve 22% of instances in 600 seconds, while formulations `conic` and `conic+` require 16 seconds and 7 seconds to solve the same quantity of instances, respectively, thus resulting in 38x and 86x speed-ups in those instances.

Table 4 presents detailed results for each dataset as a function of the budget parameter $m - h$, and Table 5 presents detailed results as a function of the regularization parameter λ. It shows for each dataset and budget/regularization parameter the average time and branch-and-bound nodes used by the solver (time-outs are counted as 600 seconds), as well as the end gaps as reported by the solver (instances solved to optimality count as 0%). We see that formulation `big-M` can solve instances if the parameter $m - h$ is small (and the number of feasible solutions $\binom{m}{m-h}$ is small as well) but struggles in other instances. Formulations `conic` and `conic+` also perform better if the parameter $m - h$ is small (since enumeration is more effective) or if the regularization parameter is large (since the relaxations are stronger). Formulation `conic` is competitive or better than `conic+` in the smaller datasets such as alcohol, but `conic+` is superior overall.

Table 4: Performance of MIO methods as a function of the budget $m - h$. Each row is an average over three instances, with different values of parameter λ. `conic+` encountered numerical difficulties in an instance of radarimage with parameter $m-h = \lfloor 0.4m \rfloor$ and $\lambda = 0.2$, which is indicated with a † in the table.

name	$m-h$	big-M			conic			conic+		
		time	nodes	gap	time	nodes	gap	time	nodes	gap
alcohol	$0.1m$	2	37,063	0.0%	1	3,880	0.0%	3	5,481	0.0%
	$0.2m$	174	2,226,198	0.0%	5	20,999	0.0%	11	34,757	0.0%
	$0.3m$	600	9,415,411	7.3%	12	40,484	0.0%	10	25,980	0.0%
	$0.4m$	576	8,681,299	7.8%	7	24,367	0.0%	14	40,210	0.0%
education	$0.1m$	4	65,793	0.0%	1	2,841	0.0%	2	1,288	0.0%
	$0.2m$	600	5,015,602	35.1%	49	73,475	0.0%	13	21,019	0.0%
	$0.3m$	600	4,037,385	60.4%	571	601,629	8.9%	143	179,170	0.0%
	$0.4m$	600	2,961,573	60.0%	600	772,419	19.6%	256	142,113	4.4%
epilepsy	$0.1m$	10	43,399	0.0%	1	1,382	0.0%	3	510	0.0%
	$0.2m$	424	4,425,843	1.1%	26	70,898	0.0%	6	9,211	0.0%
	$0.3m$	600	5,396,641	29.1%	600	1,705,862	11.1%	249	215,514	1.7%
	$0.4m$	600	6,885,941	29.8%	600	1,922,058	18.6%	351	519,278	2.1%
pulpfiber	$0.1m$	5	49,645	0.0%	2	4,385	0.0%	3	4,219	0.0%
	$0.2m$	600	2,907,040	26.5%	397	565,807	2.3%	414	591,617	0.0%
	$0.3m$	600	2,881,002	35.7%	600	839,198	13.0%	600	562,363	11.8%
	$0.4m$	600	3,161,658	36.0%	600	348,519	16.7%	600	553,325	12.5%
wagner	$0.1m$	292	3,080,940	0.0%	225	464,437	0.0%	28	68,909	0.0%
	$0.2m$	600	5,003,290	60.1%	600	873,538	36.9%	428	737,124	12.2%
	$0.3m$	600	3,249,964	65.9%	600	553,961	48.1%	600	313,817	21.4%
	$0.4m$	600	3,263,027	63.1%	600	730,974	53.0%	600	279,949	29.3%
milk	$0.1m$	600	2,166,874	52.6%	600	664,593	21.4%	424	414,886	6.8%
	$0.2m$	600	2,334,757	75.6%	600	615,562	46.7%	600	283,008	21.9%
	$0.3m$	600	2,342,248	74.7%	600	601,391	51.4%	600	263,689	26.1%
	$0.4m$	600	2,457,988	73.4%	600	453,377	52.4%	600	336,605	25.0%
foodstamp	$0.1m$	600	3,679,125	79.7%	600	962,111	15.6%	387	652,085	5.4%
	$0.2m$	600	3,428,673	94.1%	600	937,462	51.1%	600	628,270	31.0%
	$0.3m$	600	1,365,841	95.4%	600	232,394	62.7%	600	260,683	42.2%
	$0.4m$	600	1,568,131	96.2%	600	250,902	65.9%	600	260,335	47.0%
radarimage	$0.1m$	600	117,206	99.7%	600	18,268	72.7%	600	4,604	30.8%
	$0.2m$	600	219,463	99.8%	600	17,128	80.7%	600	5,066	44.1%
	$0.3m$	600	258,303	99.8%	600	21,989	87.9%	600	7,681	53.4%
	$0.4m$	600	255,181	99.9%	600	19,899	92.1%	†	†	†

5.4.4 Solution quality

We now compare the best solutions found by formulation `conic+` and heuristic `alt-opt` in the real datasets. For each instance, we compute the gap of any

Table 5: Performance of MIO methods as a function of the regularization λ. Each row is an average over four instances, with different values of parameter $m - h$. `conic+` encountered numerical difficulties in an instance of radarimage with parameter $m - h = \lfloor 0.4m \rfloor$ and $\lambda = 0.2$, which is indicated with a † in the table.

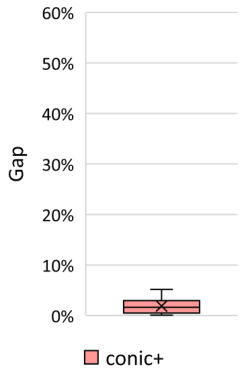
name	λ	big-M			conic			conic+		
		time	nodes	gap	time	nodes	gap	time	nodes	gap
alcohol	0.05	331	4,998,999	2.7%	15	49,984	0.0%	18	52,293	0.0%
	0.10	317	5,014,222	4.0%	4	12,738	0.0%	8	22,117	0.0%
	0.20	366	5,256,758	4.7%	1	4,576	0.0%	3	5,422	0.0%
education	0.05	451	2,672,308	32.0%	321	527,878	12.4%	244	208,764	3.3%
	0.10	452	3,013,939	38.5%	311	279,599	7.1%	51	37,059	0.0%
	0.20	451	3,374,019	46.0%	284	280,297	1.8%	15	11,871	0.0%
epilepsy	0.05	456	4,936,641	12.9%	310	733,918	9.9%	303	296,293	2.8%
	0.10	392	3,381,921	17.1%	307	998,623	7.9%	125	206,952	0.0%
	0.20	378	4,245,307	15.0%	304	1,042,610	4.5%	28	55,140	0.0%
pulpfiber	0.05	451	2,151,589	26.2%	451	513,025	11.5%	445	605,184	8.2%
	0.10	451	2,277,393	24.7%	410	432,838	6.9%	388	327,832	6.1%
	0.20	452	2,320,527	22.8%	339	372,570	5.6%	379	346,628	4.0%
wagner	0.05	533	4,175,966	49.8%	547	751,548	40.9%	467	477,785	26.0%
	0.10	524	3,663,863	47.6%	500	593,842	34.7%	453	335,998	16.0%
	0.20	512	3,108,087	44.4%	472	621,794	27.9%	322	236,066	5.1%
milk	0.05	600	2,366,186	70.0%	600	594,747	57.2%	600	276,703	36.4%
	0.10	600	2,322,270	69.8%	600	603,373	44.2%	600	395,896	19.1%
	0.20	600	2,317,945	67.3%	600	553,073	27.5%	467	301,042	4.4%
foodstamp	0.05	600	2,489,055	92.2%	600	579,845	62.9%	600	466,488	46.5%
	0.10	600	2,466,540	90.9%	600	552,682	49.4%	586	518,835	31.0%
	0.20	600	2,575,733	90.9%	600	654,626	34.2%	453	365,707	16.8%
radarimage	0.05	600	220,631	99.8%	600	22,315	92.1%	600	9,388	67.2%
	0.10	600	200,154	99.8%	600	22,358	85.5%	600	4,670	46.1%
	0.20	600	216,832	99.8%	600	13,291	72.4%	†	†	†

method as

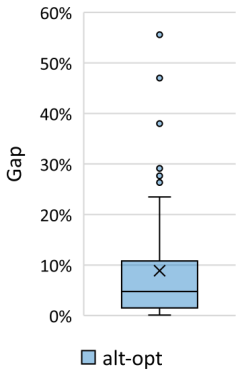
$$\text{Gap} = \frac{\zeta_{\text{method}} - \zeta^*}{\zeta^*},$$

where ζ_{method} is the objective value found by the method and ζ^* is the objective value of the best solution found for that instance (by any method). The results are presented in Figure 5. We see that `alt-opt` produced worse solutions than `conic+` in close to 40% of the instances, and in those instances the gaps are relatively large (9% on average, and has high as 50% in some instances). In contrast, `conic+` delivers worse solutions in only 5.4% of the instances, and the gaps are relatively small in those instances (2% on average). We conclude that

while `alt-opt` finds optimal solutions (or at least as good as `alt-opt`) in a good portion of the instances, it may deliver poor quality solutions when it fails. In contrast, `conic+` seems to be reliable in all cases (at the expense of additional computational time).



(a) Optimality gap of `conic+` in the 5.4% of instances where `alt-opt` produced better solutions



(b) Optimality gap of `alt-opt` in the 38.7% of instances where `conic+` produced better solutions

Figure 5: Optimality gaps of `conic+` and `alt-opt` in instances with real datasets where they are outperformed by the other method. Method `conic+` delivers optimal solutions in most of the instances, and results in small optimality gaps (average 1.9%) when outperformed. Method `alt-opt` delivers inferior solutions in more than 1/3 of the instances, with average optimality gaps of (8.9%), and larger than 25% in several instances.

6 Conclusions

We studied relaxations for a class of mixed-integer optimization problems arising often in statistics. The problems under study are characterized by products of binary variables with nonlinear quadratic terms. Few MIO approaches exist in the literature for the problems considered, and rely on big-M linearizations of the cubic terms, resulting in weak relaxations which provide trivial bounds only. In the paper, we derive the first big-M free relaxations of the problems considered, and our numerical studies with least trimmed squares instances confirm that the suggested relaxations are substantially better than the state-of-the-art. We hope that the study in the paper serves to pave the way for efficient solution of the problems considered via mixed-integer optimization.

References

[1] J. Agulló. New algorithms for computing the least trimmed squares regression estimator. *Computational Statistics & Data Analysis*, 36(4):425–439, 2001.

[2] M. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Operations Research Letters*, 37(3):187–191, 2009.

[3] A. Albert. Conditions for positive and nonnegative definiteness in terms of pseudoinverses. *SIAM Journal on Applied Mathematics*, 17(2):434–440, 1969.

[4] A. Atamtürk and A. Gómez. Rank-one convexification for sparse regression. *arXiv preprint arXiv:1901.10334*, 2019.

[5] A. Atamturk and A. Gómez. Safe screening rules for ℓ_0-regression from perspective relaxations. In *International Conference on Machine Learning*, pages 421–430. PMLR, 2020.

[6] A. Atamtürk, A. Gómez, and S. Han. Sparse and smooth signal estimation: convexification of L0-formulations. *The Journal of Machine Learning Research*, 22(1):2370–2412, 2021.

[7] W. Ben-Ameur and J. Neto. New bounds for subset selection from conic relaxations. *European Journal of Operational Research*, 298(2):425–438, 2022.

[8] T. Bernholt. Computing the least median of squares estimator in time o(nd). In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 697–706, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[9] T. Bernholt. Robust estimators are hard to compute. Technical report, Univ. Dortmund, 2005.

[10] D. Bertsimas, A. King, and R. Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

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

[12] D. Bertsimas and B. Van Parys. Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, 48(1):300–323, 2020.

[13] K. Bhatia, P. Jain, and P. Kar. Robust regression via hard thresholding. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 721–729, Cambridge, MA, USA, 2015. MIT Press.

[14] A. Cozad, N. V. Sahinidis, and D. C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

[15] A. Cozad, N. V. Sahinidis, and D. C. Miller. A combined first-principles and data-driven approach to model building. *Computers & Chemical Engineering*, 73:116–127, 2015.

[16] H. Dong, K. Chen, and J. Linderoth. Regularization vs. relaxation: A conic optimization perspective of statistical variable selection. *arXiv preprint arXiv:1510.06083*, 2015.

[17] J. W. Dunn. *Optimal trees for prediction and prescription.* PhD thesis, Massachusetts Institute of Technology, 2018.

[18] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on matrix analysis and applications*, 18(4):1035–1064, 1997.

[19] A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. *Mathematical Programming*, 106:225–236, 2006.

[20] A. Giloni and M. Padberg. Least trimmed squares regression, least median squares regression, and mathematical programming. *Mathematical and Computer Modelling*, 35(9-10):1043–1060, 2002.

[21] A. Gómez. Outlier detection in time series via mixed-integer conic quadratic optimization. *SIAM Journal on Optimization*, 31(3):1897–1925, 2021.

[22] A. Gómez and O. A. Prokopyev. A mixed-integer fractional optimization approach to best subset selection. *INFORMS Journal on Computing*, 33(2):551–565, 2021.

[23] O. Günlük and J. Linderoth. Perspective reformulations of mixed integer nonlinear programs with indicator variables. *Mathematical programming*, 124:183–205, 2010.

[24] F. R. Hampel. A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, 42(6):1887–1896, 1971.

[25] D. M. Hawkins. The feasible solution algorithm for least trimmed squares regression. *Computational Statistics & Data Analysis*, 17(2):185–196, 1994.

[26] H. Hazimeh and R. Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, 68(5):1517–1537, 2020.

[27] H. Hazimeh, R. Mazumder, and T. Nonet. L0Learn: A scalable package for sparse learning using L0 regularization. *arXiv preprint arXiv:2202.04820*, 2022.

[28] H. Hazimeh, R. Mazumder, and A. Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, 196(1-2):347–388, 2022.

[29] P. Huber. Robust regression: Asymptotics, conjectures and monte carlo. *Ann. Statist.*, 1:799–821, 1973.

[30] P. Huber. *Robust Statistics*. Springer, Berlin, 2011.

[31] L. Insolia, A. Kenney, F. Chiaromonte, and G. Felici. Simultaneous feature selection and outlier detection with optimality guarantees. *Biometrics*, 78(4):1592–1603, 2022.

[32] K. Kimura and H. Waki. Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program. *Optimization Methods and Software*, 33(3):633–649, 2018.

[33] J. Kronqvist, R. Misener, and C. Tsay. P-split formulations: A class of intermediate formulations between big-M and convex hull for disjunctive constraints. *arXiv preprint arXiv:2202.05198*, 2022.

[34] P. Liu, S. Fattahi, A. Gómez, and S. Küçükyavuz. A graph-based decomposition method for convex quadratic optimization with indicators. *Mathematical Programming*, pages 1–33, 2022.

[35] T.-T. Lu and S.-H. Shiou. Inverses of 2×2 block matrices. *Computers & Mathematics with Applications*, 43(1):119–129, 2002.

[36] H. Manzour, S. Küçükyavuz, H.-H. Wu, and A. Shojaie. Integer programming for learning directed acyclic graphs from continuous data. *INFORMS Journal on Optimization*, 3(1):46–73, 2021.

[37] R. Mazumder, P. Radchenko, and A. Dedieu. Subset selection with shrinkage: Sparse linear modeling when the snr is low. *Operations Research*, 71(1):129–147, 2023.

[38] R. Mazumder and H. Wang. Linear regression with partially mismatched data: local search with theoretical guarantees. *Mathematical Programming*, 197(2):1265–1303, 2023.

[39] R. Miyashiro and Y. Takano. Mixed integer second-order cone programming formulations for variable selection in linear regression. *European Journal of Operational Research*, 247(3):721–731, 2015.

[40] D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu. On the least trimmed squares estimator. *Algorithmica*, 69, 2014.

[41] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[42] Y. W. Park and D. Klabjan. Subset selection for multiple linear regression via optimization. *Journal of Global Optimization*, 77(3):543–574, 2020.

[43] P. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[44] P. Rousseeuw, C. Croux, V. Todorov, A. Ruckstuhl, M. Salibian-Barrera, T. Verbeke, M. Koller, and M. Maechler. Robustbase: basic robust statistics. *R package version 0.4-5, URL http://CRAN. R-project. org/package= robustbase*, 2009.

[45] P. Rousseeuw and A. Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 1987.

[46] P. Rousseeuw and K. Van Driessen. Computing LTS regression for large data sets. *Data Min Knowl Disc*, 12:29–45, 2006.

[47] P. Rousseeuw and V. Yohai. Robust regression by means of s-estimators. In *Robust and Nonlinear Time Series Analysis: Proceedings of a Workshop Organized by the Sonderforschungsbereich 123 "Stochastische Mathematische Modelle", Heidelberg 1983*, pages 256–272. Springer, 1984.

[48] Y. Shen and S. Sanghavi. Iterative least trimmed squares for mixed linear regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[49] Y. Shen and S. Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5739–5748. PMLR, 09–15 Jun 2019.

[50] J. Sherman and W. Morrison. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Annals of Mathematical Statistics*, 20(4):620–624, 1949.

[51] J. Sherman and W. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[52] L. Wei, A. Atamtürk, A. Gómez, and S. Küçükyavuz. On the convex hull of convex quadratic optimization problems with indicators. *Forthcoming in Mathematical Programming*, 2023.

[53] Z. T. Wilson and N. V. Sahinidis. The alamo approach to machine learning. *Computers & Chemical Engineering*, 106:785–795, 2017.

[54] W. Xie and X. Deng. Scalable algorithms for the sparse ridge regression. *SIAM Journal on Optimization*, 30(4):3359–3386, 2020.

[55] X. Zheng, X. Sun, and D. Li. Improving the performance of MIQP solvers for quadratic programs with cardinality and minimum threshold constraints: A semidefinite program approach. *INFORMS Journal on Computing*, 26(4):690–703, 2014.

[56] G. Zioutas and A. Avramidis. Deleting outliers in robust regression with mixed integer programming. *Acta Mathematicae Applicatae Sinica*, 21:323–334, 2005.

[57] G. Zioutas, L. Pitsoulis, and A. Avramidis. Quadratic mixed integer programming and support vectors for deleting outliers in robust regression. *Annals of Operations Research*, 166(1):339–353, 2009.