

# LEAST QUANTILE REGRESSION VIA MODERN OPTIMIZATION

BY DIMITRIS BERTSIMAS\* AND RAHUL MAZUMDER†

*Massachusetts Institute of Technology\* and Columbia University†*

We address the Least Quantile of Squares (LQS) (and in particular the Least Median of Squares) regression problem using modern optimization methods. We propose a Mixed Integer Optimization (MIO) formulation of the LQS problem which allows us to find a provably globally optimal solution for the LQS problem. Our MIO framework has the appealing characteristic that if we terminate the algorithm early, we obtain a solution with a guarantee on its sub-optimality. We also propose continuous optimization methods based on first order subgradient descent, sequential linear optimization and hybrid combinations of them to obtain near optimal solutions to the LQS problem. The MIO algorithm is found to benefit significantly from high quality solutions delivered by our continuous optimization based methods. We further show that the MIO approach leads to **(a)** an optimal solution for *any* dataset, where the data-points  $(y_i, \mathbf{x}_i)$ 's are not necessarily in general position, **(b)** a simple proof of the breakdown point of the LQS objective value that holds for any dataset and **(c)** an extension to situations where there are polyhedral constraints on the regression coefficient vector. We report computational results with both synthetic and real-world datasets showing that the MIO algorithm with warm starts from the continuous optimization methods solve small ( $n = 100$ ) and medium ( $n = 500$ ) size problems to provable optimality in under two hours, and outperform all publicly available methods for large scale ( $n = 10,000$ ) LQS problems.

**1. Introduction.** Consider a linear model with response  $\mathbf{y} \in \mathfrak{R}^n$  model matrix  $\mathbf{X}_{n \times p}$ , regression coefficients  $\boldsymbol{\beta} \in \mathfrak{R}^p$  and error  $\boldsymbol{\epsilon} \in \mathfrak{R}^n$ :

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

We will assume that  $\mathbf{X}$  contains a column of ones to account for the intercept in the model. Given data for the  $i$ th sample  $(y_i, \mathbf{x}_i)$ ,  $i = 1, \dots, n$  (where,  $\mathbf{x}_i \in \mathfrak{R}^{p \times 1}$ ) and regression coefficients  $\boldsymbol{\beta}$ , the  $i$ th residual is given by the usual notation  $r_i = y_i - \mathbf{x}_i' \boldsymbol{\beta}$  for  $i = 1, \dots, n$ . The traditional Least Squares (LS) estimator given by

$$(1.1) \quad \hat{\boldsymbol{\beta}}^{(\text{LS})} \in \arg \min_{\boldsymbol{\beta}} \sum_{i=1}^n r_i^2$$

is a popular and effective method for estimating the regression coefficients when the error vector  $\boldsymbol{\epsilon}$  has *small*  $\ell_2$ -norm. However, in the presence of outliers, the LS estimators do not work favorably—a single outlier can have an arbitrarily large effect on the estimate. The robustness of an estimator vis-a-vis outliers is often quantified by the notion of its finite sample breakdown point ([Donoho](#)

---

*MSC 2010 subject classifications:* Primary 62J05,62G35; secondary 90C11,90C26

*Keywords and phrases:* Least Median of Squares, Robust Statistics, Least Quantile Regression, Algorithms, Mixed Integer Programming, Global Optimization, Continuous Optimization

and Huber, 1983; Hampel, 1971). The LS estimate (1.1) has a limiting (in the limit  $n \rightarrow \infty$  with  $p$  fixed) breakdown point (Hampel, 1971) of zero.

The Least Absolute Deviation (LAD) estimator given by:

$$(1.2) \quad \hat{\beta}^{(\text{LAD})} \in \arg \min_{\beta} \sum_{i=1}^n |r_i|$$

considers the  $\ell_1$ -norm on the residuals, thereby implicitly assuming that the error vector  $\epsilon$  has small  $\ell_1$ -norm. The LAD estimator is not resistant to large deviations in the covariates and, like the optimal LS solutions, has a breakdown point of zero (in the limit  $n \rightarrow \infty$  with  $p$  fixed).

M-Estimators (Huber, 1973) are obtained by minimizing a loss function of the residuals of the form  $\sum_{i=1}^n \rho(r_i)$ , where  $\rho(r)$  is a symmetric function with a unique minimum at zero. Examples include the Huber function and the Tukey function (Rousseeuw and Leroy, 1987; Huber, 2011), among others. M-estimators often simultaneously estimate the scale parameter along with the regression coefficient. M-estimators too are severely affected by the presence of outliers in the covariate space. A generalization of M-Estimators are Generalized M-Estimators (Rousseeuw and Leroy, 1987; Huber, 2011), which bound the influence of outliers in the covariate space by the choice of a weight function dampening the effect of outlying covariates. In some cases, they have an improved finite-sample breakdown point of  $1/(p+1)$ .

The repeated median estimator (Siegel, 1982) with breakdown point of approximately 50%, was one of the earliest estimators to achieve a very high breakdown point. The estimator however, is not equivariant under linear transformations of the covariates.

Rousseeuw (1984) introduced Least Median of Squares (LMS) (Hampel, 1975, see also) which minimizes the median of the absolute residuals<sup>1</sup>

$$(1.3) \quad \hat{\beta}^{(\text{LMS})} \in \arg \min_{\beta} \left( \text{median}_{i=1, \dots, n} |r_i| \right).$$

The LMS problem is equivariant and has a limiting breakdown point of 50% — making it the first equivariant estimator to achieve the maximal possible breakdown point in the limit  $n \rightarrow \infty$  with  $p$  fixed.

Instead of considering the median, one may consider more generally, the  $q^{\text{th}}$  order statistic, which leads to the Least Quantile of Squares (LQS) estimator:

$$(1.4) \quad \hat{\beta}^{(\text{LQS})} \in \arg \min_{\beta} |r_{(q)}|,$$

where  $r_{(q)}$  denotes the residual, corresponding to the  $q$ th ordered absolute residual:

$$(1.5) \quad |r_{(1)}| \leq |r_{(2)}| \leq \dots \leq |r_{(n)}|.$$

Rousseeuw (1984) showed that if the sample points  $(y_i, \mathbf{x}_i), i = 1, \dots, n$  are in general position, i.e., for any subset of  $\mathcal{I} \subset \{1, \dots, n\}$  with  $|\mathcal{I}| = p$ , the  $p \times p$  sub-matrix  $X_{\mathcal{I}}$  has rank  $p$ ; an optimal LMS solution (1.3) exists and has a finite sample breakdown point of  $(\lfloor n/2 \rfloor - p + 2)/n$ , where  $\lfloor s \rfloor$  denotes the largest integer smaller than or equal to  $s$ . Rousseeuw (1984) showed that the finite sample breakdown point of the estimator (1.3) can be further improved to achieve the maximum possible

---

<sup>1</sup>Note that the original definition of LMS (Rousseeuw, 1984) considers the squared residuals instead of the absolute values. However, we will work with the absolute values, since the problems are equivalent.

finite sample breakdown point if one considers the estimator (1.4) with  $q = \lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$ . The LMS estimator has low efficiency (Rousseeuw, 1984). This can, however, be improved by using certain post-processing methods on the LMS estimator — the one step M-estimator of Bickel (1975) or a reweighted least-squares estimator, where points with large values of LMS residuals are given small weight are popular methods that are used in this vein.

*Related work.* It is a well recognized fact that the LMS problem is computationally demanding due to the combinatorial nature of the problem. Bernholt (2005a) showed that computing an optimal LMS solution is NP-hard.

Many algorithms based on different approaches have been proposed for the LMS problem over the past thirty years. State of the art algorithms, however, fail to obtain a global minimum of the LMS problem for problem sizes larger than  $n = 50, p = 5$ . This severely limits the use of LMS for important real world multivariate applications, where  $n$  can easily range in the order of a few thousands. It goes without saying that a poor local minimum for the LMS problem may be misleading from a statistical inference point of view (see also Stromberg (1993) and references therein for related discussions on this matter). The various algorithms presented in the literature for the LMS can be placed into two very broad categories. One approach computes an optimal solution to the LMS problem using geometric characterizations of the fit — they typically rely on complete enumeration and have complexity  $O(n^p)$ . The other approach gives up on obtaining an optimal solution and resorts to heuristics and/or randomized algorithms to obtain approximate solutions to the LMS problem. These methods, to the best of our knowledge, do not provide certificates about the quality of the solution obtained. We describe below a brief overview of existing algorithms for LMS.

Among the various algorithms proposed in the literature, for the LMS problem, the most popular seems to be PROGRESS (Program for Robust Regression) (Rousseeuw and Leroy, 1987; Rousseeuw and Hubert, 1997). The algorithm does a complete enumeration of all  $p$ -subsets of the  $n$  sample points, computes the hyperplane passing through them and finds the configuration leading to the smallest value of the objective. The algorithm has a runtime complexity of  $O(n^p)$  and assumes that the data points are in general position. For computational scalability, heuristics that randomly sample subsets are often used. See also Barreto and Maharry (2006) for a recent work on algorithms for the bivariate regression problem.

Steele and Steiger (1986) proposed exact algorithms for LMS for  $p = 2$  with complexity  $O(n^3)$  and some probabilistic speed-up methods with complexity  $O((n \log(n))^2)$ .

Stromberg (1993) proposed an exact algorithm for LMS with runtime  $O(n^{(p+2)} \log(n))$  using some insightful geometric properties of the LMS fit. This method does a brute force search among  $\binom{n}{p+1}$  different regression coefficient values and scales up to problem sizes  $n = 50$  and  $p = 5$ .

Agullo (1997) proposed a finite branch and bound technique with run-time complexity  $O(n^{p+2})$  to obtain an optimal solution to the LMS problem motivated by the work of Stromberg (1993). The algorithm showed superior performance compared to methods preceding it and can scale up to problem sizes  $n \approx 70, p \approx 4$ .

Erickson, Har-Peled and Mount (2006) gives an exact algorithm with runtime  $O(n^p \log(n))$  for LMS and also show that computing an optimal LMS solution requires  $O(n^p)$  time. For the low dimensional case  $p = 2$ , Souvaine and Steele (1987) proposed an exact algorithm for LMS with complexity  $O(n^2)$  using the topological sweep-line technique.

Giloni and Padberg (2002) propose integer optimization formulations for the LMS problem, however, no computational experiments are reported—the practical performance of the proposed method thus remains unclear.

Mount et al. (2007) present an algorithm based on branch and bound for  $p = 2$  for computing approximate solutions to the LMS problem. Mount et al. (2000) present a quantile approximation algorithm with approximation factor  $\epsilon$  with complexity  $O(n \log(n) + (1/\epsilon)^{O(p)})$ . Chakraborty and Chaudhuri (2008) present probabilistic search algorithms for a class of problems in robust statistics. Nunkesser and Morell (2010) describe computational procedures based on heuristic search strategies using evolutionary algorithms for some robust statistical estimation problems including LMS. Hawkins (1993) propose a probabilistic algorithm for LMS known as the “Feasible Set Algorithm” capable of solving problems up to sizes  $n = 100, p = 3$ .

Bernholt (2005b) describe a randomized algorithm for computing the LMS running in  $O(n^p)$  time and  $O(n)$  space, for fixed  $p$ . Olson (1997) describes an approximation algorithm to compute an optimal LMS solution within an approximation factor of two using randomized sampling methods—the method has (expected) run-time complexity of  $O(n^{p-1} \log(n))$ .

*Related Approaches in Robust Regression.* Other estimation procedures that achieve a high breakdown point and good statistical efficiency include the least trimmed squares estimator (Rousseeuw, 1984; Rousseeuw and Leroy, 1987), which minimizes the sum of squares of the  $q$  smallest squared residuals. Another popular approach are based on S-estimators (Rousseeuw, 1984; Rousseeuw and Leroy, 1987), which are a type of M-estimators of scale on the residuals. These estimation procedures like the LMS estimator are NP-hard (Bernholt, 2005a).

We refer the interested reader to Hubert, Rousseeuw and Van Aelst (2008) for a nice review of various robust statistical methods and their applications (Meer et al., 1991; Stewart, 1999; Rousseeuw et al., 2006).

In this paper, we propose a computationally tractable framework to compute a globally optimal solution to the LQS Problem (1.4), and in particular the LMS problem via modern optimization methods: first order methods from continuous optimization and mixed integer optimization (MIO), see Bertsimas and Weismantel (2005). Our view of computational tractability is not polynomial time solution times as these do not exist for the LQS problem unless  $P=NP$ . Rather it is the ability of a method to solve problems of practical interest in times that are appropriate for the application addressed. An important advantage of our framework is that it easily adapts to obtain solutions to more general variants of (1.4) under polyhedral constraints, i.e.,

$$(1.6) \quad \underset{\boldsymbol{\beta}}{\text{minimize}} \quad |r_{(q)}| \quad \text{subject to} \quad \mathbf{A}\boldsymbol{\beta} \leq \mathbf{b},$$

where  $\mathbf{A}_{m \times p}, \mathbf{b}_{m \times 1}$  are given parameters in the problem representing side constraints on the variable  $\boldsymbol{\beta}$  and “ $\leq$ ” denotes component wise inequality. This is useful if one would like to incorporate some form of regularization on the  $\boldsymbol{\beta}$  coefficients, for example:  $\ell_1$  regularization (Tibshirani, 1996) or a generalized<sup>2</sup>  $\ell_1$  regularization on  $\boldsymbol{\beta}$  (Tibshirani and Taylor, 2011).

*Contributions.* Our contributions in this paper may be summarized as follows:

1. We use MIO to find a provably optimal solution to the LQS problem. Our framework has the appealing characteristic that if we terminate the algorithm early, we obtain a solution with a guarantee on its suboptimality. We further show that the MIO approach leads to an optimal solution for *any* dataset where the data-points  $(y_i, \mathbf{x}_i)$ ’s are not necessarily in general

---

<sup>2</sup>A generalized  $\ell_1$  regularization on the regression coefficients is given by a constraint set of the form  $\|D\boldsymbol{\beta}\|_1 \leq \lambda$  for some given matrix  $D_{m \times p}$  and a shrinkage/sparsity level  $\lambda$ .

position. The MIO formulation enables us to provide a simple proof of the breakdown point of the LQS objective value, generalizing the existing results for the problem. Furthermore, our approach is readily generalizable to problems of the type (1.6).

2. We introduce a variety of solution methods based on modern continuous optimization — first order subgradient descent, sequential linear optimization and a hybrid version of these two methods that provide near optimal solutions for the LQS problem. The MIO algorithm is found to significantly benefit from solutions obtained by the continuous optimization methods.
3. We report computational results with both synthetic and real-world datasets that show that the MIO algorithm with warm starts from the continuous optimization methods solve small ( $n = 100$ ) and medium ( $n = 500$ ) size LQS problems to provable optimality in under two hours, and outperform all publicly available methods for large ( $n = 10,000$ ) scale LQS problems, but without showing provable optimality in under two hours of computation.

*Structure of the paper.* The paper is organized as follows. Section 2 describes MIO approaches for the LQS problem. Section 3 describes continuous optimization based methods for obtaining local minimizers for the LQS problem. Section 4 describes properties of an optimal LQS solution. Section 5 describes computational results and experiments. The last section contains our key conclusions.

**2. Mixed Integer Optimization Formulation.** In this section, we present an exact MIO formulation for the LQS problem. For the sake of completeness, we will first introduce the definition of a linear MIO problem. The generic MIO framework concerns the following optimization problem:

$$\begin{aligned}
 (2.1) \quad & \text{minimize} && \mathbf{c}'\boldsymbol{\alpha} + \mathbf{d}'\boldsymbol{\theta} \\
 & && A\boldsymbol{\alpha} + B\boldsymbol{\theta} \geq \mathbf{b} \\
 & && \boldsymbol{\alpha} \in \mathfrak{R}_+^n \\
 & && \boldsymbol{\theta} \in \{0, 1\}^m,
 \end{aligned}$$

where  $\mathbf{c} \in \mathfrak{R}^n$ ,  $\mathbf{d} \in \mathfrak{R}^m$ ,  $A \in \mathfrak{R}^{k \times n}$ ,  $B \in \mathfrak{R}^{k \times m}$ ,  $\mathbf{b} \in \mathfrak{R}^k$  are the given parameters of the problem;  $\mathfrak{R}_+^n$  denotes the non-negative  $n$ -dimensional orthant, the symbol  $\geq$  denotes element-wise inequalities and we optimize over both continuous ( $\boldsymbol{\alpha}$ ) and discrete ( $\boldsymbol{\theta}$ ) variables. For background on MIO see [Bertsimas and Weismantel \(2005\)](#).

Consider a list of  $n$  numbers  $|r_1|, \dots, |r_n|$ , with the ordering described in (1.5). To model the sorted  $q$ -th residual, i.e.,  $|r_{(q)}|$ , we need to express the fact that  $r_i \leq |r_{(q)}|$  for  $q$  many residuals  $|r_i|$ 's from  $|r_1|, \dots, |r_n|$ . To do so we introduce the binary variables  $z_i, i = 1, \dots, n$  with the interpretation:

$$(2.2) \quad z_i = \begin{cases} 1, & \text{if } |r_i| \leq |r_{(q)}|, \\ 0, & \text{otherwise.} \end{cases}$$

We further introduce auxiliary continuous variables  $\mu_i, \bar{\mu}_i \geq 0$  such that:

$$(2.3) \quad |r_i| - \mu_i \leq |r_{(q)}| \leq |r_i| + \bar{\mu}_i, i = 1, \dots, n,$$

with the conditions:

$$(2.4) \quad \begin{aligned} & \text{If } |r_i| \geq |r_{(q)}|, \text{ then } \bar{\mu}_i = 0, \mu_i \geq 0, \\ & \text{and if } |r_i| \leq |r_{(q)}|, \text{ then } \mu_i = 0, \bar{\mu}_i \geq 0. \end{aligned}$$

We thus propose the following MIO formulation:

$$\begin{aligned}
(2.5) \quad & \text{minimize} && \gamma \\
& \text{subject to} && |r_i| + \bar{\mu}_i \geq \gamma, \quad i = 1, \dots, n \\
& && \gamma \geq |r_i| - \mu_i, \quad i = 1, \dots, n \\
& && M_u z_i \geq \bar{\mu}_i, \quad i = 1, \dots, n \\
& && M_\ell (1 - z_i) \geq \mu_i, \quad i = 1, \dots, n \\
& && \sum_{i=1}^n z_i = q \\
& && \mu_i \geq 0, \quad i = 1, \dots, n \\
& && \bar{\mu}_i \geq 0, \quad i = 1, \dots, n \\
& && z_i \in \{0, 1\}, \quad i = 1, \dots, n,
\end{aligned}$$

where,  $\gamma, z_i, \mu_i, \bar{\mu}_i, i = 1, \dots, n$  are the optimization variables,  $M_u, M_\ell$  are sufficiently large constants. Let us denote the optimal solution of problem (2.5), which depends on  $M_\ell, M_u$ , by  $\gamma^*$ . Suppose we consider  $M_u, M_\ell \geq \max_i |r_{(i)}|$ —it follows from formulation (2.5) that  $q$  of the  $\mu_i$ 's are zero. Thus,  $\gamma^*$  has to be larger than at least  $q$  of the  $|r_i|$  values. By arguments similar to the above, we see that, since  $(n - q)$  of the  $z_i$ 's are zero, at least  $(n - q)$  many  $\bar{\mu}_i$ 's are zero. Thus  $\gamma^*$  is less than or equal to at least  $(n - q)$  many of the  $|r_i|, i = 1, \dots, n$  values. This shows that  $\gamma^*$  is indeed equal to  $|r_{(q)}|$ , for  $M_u, M_\ell$  sufficiently large.

We found in our experiments that, in formulation (2.5), if  $z_i = 1$ , then  $\bar{\mu}_i = M_u$  and if  $z_i = 0$  then  $\mu_i = M_\ell$ . Though this does not interfere with the definition of  $|r_{(q)}|$ , it creates a difference in the strength of the MIO formulation. We describe below how to circumvent this shortcoming.

From (2.4) it is clear that  $\bar{\mu}_i \mu_i = 0, \forall i = 1, \dots, n$ . The constraint  $\bar{\mu}_i \mu_i = 0$  can be modeled via integer optimization using Specially Ordered Sets of Type 1 (Bertsimas and Weismantel, 2005), i.e., SOS-1 constraints as follows:

$$(2.6) \quad \mu_i \bar{\mu}_i = 0 \iff (\mu_i, \bar{\mu}_i) : \text{SOS-1},$$

for every  $i = 1, \dots, n$ . In addition, observe that, for  $M_\ell$  sufficiently large and every  $i \in \{1, \dots, n\}$  the constraint  $M_\ell (1 - z_i) \geq \mu_i \geq 0$  can be modeled<sup>3</sup> by a SOS-1 constraint —  $(\mu_i, z_i) : \text{SOS-1}$ . In light of this discussion, we see that

$$(2.7) \quad |r_i| - |r_{(q)}| = \mu_i - \bar{\mu}_i, \quad (\mu_i, \bar{\mu}_i) : \text{SOS-1}.$$

We next show that  $|r_{(q)}| \geq \bar{\mu}_i$  and  $\mu_i \leq |r_i|$  for all  $i = 1, \dots, p$ . When  $|r_i| \leq |r_{(q)}|$  it follows from the above representation that

$$\mu_i = 0 \text{ and } \bar{\mu}_i = |r_{(q)}| - |r_i| \leq |r_{(q)}|.$$

When  $|r_i| > |r_{(q)}|$ , it follows that  $\bar{\mu}_i = 0$ . Thus, it follows that  $0 \leq \bar{\mu}_i \leq |r_{(q)}|$  for all  $i = 1, \dots, n$ . It also follows by a similar argument that  $0 \leq \mu_i \leq |r_i|$  for all  $i$ .

<sup>3</sup>To see why this is true observe that  $(\mu_i, z_i) : \text{SOS-1}$  is equivalent to  $\mu_i z_i = 0$ . Now, since  $z_i \in \{0, 1\}$ , we have the following possibilities:  $z_i = 0$ , in which case  $\mu_i$  is free; if  $z_i = 1$ , then  $\mu_i = 0$ .

Thus, by using SOS-1 type of constraints, we can avoid the use of Big-M's appearing in formulation (2.5), as follows:

$$\begin{aligned}
(2.8) \quad & \text{minimize} && \gamma \\
& \text{subject to} && |r_i| - \gamma = \mu_i - \bar{\mu}_i, \quad i = 1, \dots, n \\
& && \sum_{i=1}^n z_i = q \\
& && \gamma \geq \bar{\mu}_i, \quad i = 1, \dots, n \\
& && \bar{\mu}_i \geq 0, \quad i = 1, \dots, n \\
& && \mu_i \geq 0, \quad i = 1, \dots, n \\
& && (\bar{\mu}_i, \mu_i) : \text{SOS-1}, \quad i = 1, \dots, n \\
& && (z_i, \mu_i) : \text{SOS-1}, \quad i = 1, \dots, n \\
& && z_i \in \{0, 1\}, \quad i = 1, \dots, n.
\end{aligned}$$

Note, however, that the constraints

$$(2.9) \quad |r_i| - \gamma = \mu_i - \bar{\mu}_i, \quad i = 1, \dots, n$$

are not convex in  $r_1, \dots, r_n$ . We thus introduce the following variables  $r_i^+, r_i^-, i = 1, \dots, n$  such that

$$(2.10) \quad r_i^+ + r_i^- = |r_i|, \quad y_i - \mathbf{x}_i' \boldsymbol{\beta} = r_i^+ - r_i^-, \quad r_i^+ \geq 0, r_i^- \geq 0, \quad r_i^+ r_i^- = 0, \quad i = 1, \dots, n.$$

The constraint  $r_i^+ r_i^- = 0$  can be modeled via SOS-1 constraints

$$(r_i^+, r_i^-) : \text{SOS-1 for every } i = 1, \dots, n.$$

This leads to the following MIO for the LQS problem that we use in this paper:

$$\begin{aligned}
(2.11) \quad & \text{minimize} && \gamma \\
& \text{subject to} && r_i^+ + r_i^- - \gamma = \bar{\mu}_i - \mu_i, \quad i = 1, \dots, n \\
& && r_i^+ - r_i^- = y_i - \mathbf{x}_i' \boldsymbol{\beta}, \quad i = 1, \dots, n \\
& && \sum_{i=1}^n z_i = q \\
& && \gamma \geq \mu_i \geq 0, \quad i = 1, \dots, n \\
& && \mu_i \geq 0, \quad i = 1, \dots, n \\
& && \bar{\mu}_i \geq 0, \quad i = 1, \dots, n \\
& && r_i^+ \geq 0, r_i^- \geq 0, \quad i = 1, \dots, n \\
& && (\bar{\mu}_i, \mu_i) : \text{SOS-1}, \quad i = 1, \dots, n \\
& && (r_i^+, r_i^-) : \text{SOS-1}, \quad i = 1, \dots, n \\
& && (z_i, \mu_i) : \text{SOS-1}, \quad i = 1, \dots, n \\
& && z_i \in \{0, 1\}, \quad i = 1, \dots, n.
\end{aligned}$$

To motivate the reader, we show in Figure 1 an example that illustrates that the MIO formulation (2.11) leads to a provably optimal solution for the LQS problem. We give more details in Section 5.

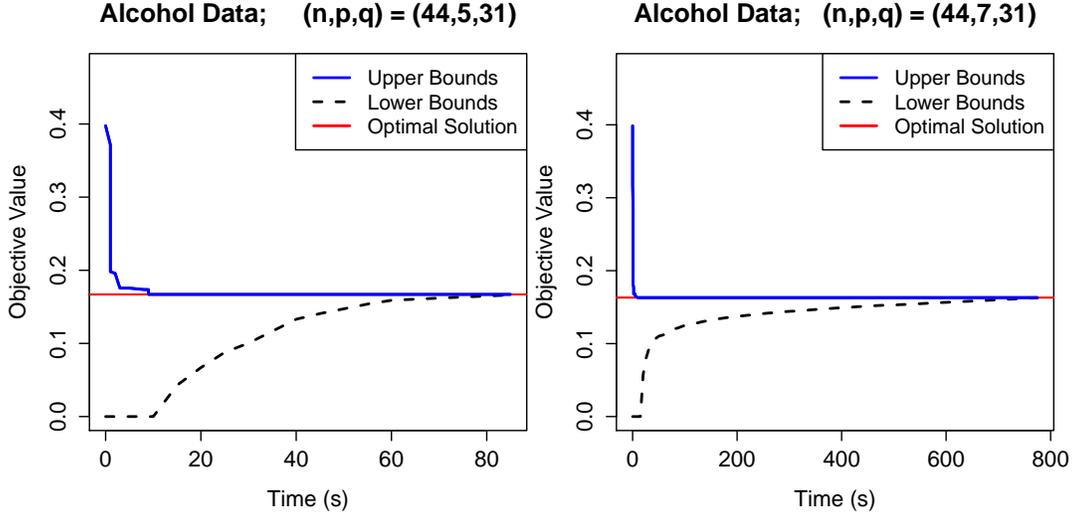


FIG 1. Figure showing the typical evolution of the MIO formulation (2.11) for the “Alcohol” dataset with  $n = 44, q = 31$  with  $p = 5$  (left panel) and  $p = 7$  (right panel). Global solutions for both the problems are found quite quickly in both examples, but it takes longer to certify global optimality via the lower bounds. As expected, the time taken for the MIO to certify convergence to the global optimum increases with increasing  $p$ .

**3. Continuous Optimization Based Methods.** We describe two main approaches based on continuous optimization for the LQS problem. Section 3.1 presents a method based on sequential linear optimization and Section 3.2 describes a first order sub-gradient based method for the LQS problem. Section 3.3 describes hybrid combinations of the aforementioned approaches, which we have found, empirically, to provide high quality solutions. Section 3.4 describes initialization strategies for the algorithms.

**3.1. Sequential Linear Optimization.** We describe a sequential linear optimization approach to obtain a local minimum of Problem (1.4). We first describe the algorithm, present its convergence analysis and describe its iteration complexity.

*Main description of the Algorithm.* We decompose the  $q$ th ordered absolute residual as follows:

$$(3.1) \quad |r_{(q)}| = |y_{(q)} - \mathbf{x}'_{(q)}\boldsymbol{\beta}| = \underbrace{\sum_{i=1}^{q+1} |y_{(i)} - \mathbf{x}'_{(i)}\boldsymbol{\beta}|}_{H_{q+1}(\boldsymbol{\beta})} - \underbrace{\sum_{i=1}^q |y_{(i)} - \mathbf{x}'_{(i)}\boldsymbol{\beta}|}_{H_q(\boldsymbol{\beta})},$$

The function  $H_m(\boldsymbol{\beta})$  can be written as

$$(3.2) \quad \begin{aligned} H_m(\boldsymbol{\beta}) := & \max_{\mathbf{w}} \sum_{i=1}^n w_i |y_i - \mathbf{x}'_i \boldsymbol{\beta}| \\ & \text{subject to } \sum_{i=1}^n w_i = m \\ & 0 \leq w_i \leq 1, \quad i = 1, \dots, n. \end{aligned}$$

Let us denote the feasible set in problem (3.2) by

$$\mathcal{W}_m := \left\{ \mathbf{w} : \sum_{i=1}^n w_i = m, w_i \in [0, 1], \forall i = 1, \dots, n \right\}.$$

Observe that for every  $\mathbf{w} \in \mathcal{W}_m$  the function  $\sum_{i=1}^n w_i |y_i - \mathbf{x}'_i \boldsymbol{\beta}|$  is convex in  $\boldsymbol{\beta}$ . Furthermore, since  $H_m(\boldsymbol{\beta})$  is the point-wise supremum with respect to  $\mathbf{w}$  over  $\mathcal{W}_m$ , the function  $H_m(\boldsymbol{\beta})$  is convex in  $\boldsymbol{\beta}$  (see [Boyd and Vandenberghe \(2004\)](#)). By taking the dual of Problem (3.2) for  $m = q + 1$  and invoking strong duality, we have:

$$(3.3) \quad \begin{aligned} H_{q+1}(\boldsymbol{\beta}) = & \min_{\theta, \boldsymbol{\nu}} \theta (q + 1) + \sum_{i=1}^n \nu_i \\ & \text{subject to } \theta + \nu_i \geq |y_i - \mathbf{x}'_i \boldsymbol{\beta}|, \quad i = 1, \dots, n \\ & \nu_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

Hence, we have expressed the  $q$ th ordered absolute residual as a difference of two convex functions. Having expressed  $H_{q+1}(\boldsymbol{\beta})$  as the value of a LO problem we linearize the function  $H_q(\boldsymbol{\beta})$ . Representation (3.2) also provides a characterization of the set of subgradients of  $H_q(\boldsymbol{\beta})$ :

$$(3.4) \quad \partial H_q(\boldsymbol{\beta}) = \text{conv} \left\{ \sum_{i=1}^n w_i^* \text{sgn}(y_i - \mathbf{x}'_i \boldsymbol{\beta}) \mathbf{x}_i : \mathbf{w}^* \in \arg \max_{\mathbf{w} \in \mathcal{W}_q} \mathcal{L}(\boldsymbol{\beta}, \mathbf{w}) \right\},$$

where  $\mathcal{L}(\boldsymbol{\beta}, \mathbf{w}) = \sum_{i=1}^n w_i |y_i - \mathbf{x}'_i \boldsymbol{\beta}|$  and ‘conv’( $S$ )’ denotes the convex hull of set  $S$ . An element of the set of subgradients (3.4) will be denoted by  $\partial H(\boldsymbol{\beta})$ .

If  $\boldsymbol{\beta}_k$  denotes the value of the estimate at iteration  $k$ , we linearize  $H_q(\boldsymbol{\beta})$  at  $\boldsymbol{\beta}_k$  as follows:

$$(3.5) \quad H_q(\boldsymbol{\beta}) \approx H_q(\boldsymbol{\beta}_k) + \langle \partial H_q(\boldsymbol{\beta}_k), \boldsymbol{\beta} - \boldsymbol{\beta}_k \rangle.$$

Combining (3.3) and (3.5) we obtain that minimizing (3.1) with respect to  $\boldsymbol{\beta}$  can be approximately done by solving the following LO problem:

$$(3.6) \quad \begin{aligned} \min_{\nu, \theta, \boldsymbol{\beta}} & \theta (q + 1) + \sum_{i=1}^n \nu_i - \langle \partial H_q(\boldsymbol{\beta}_k), \boldsymbol{\beta} \rangle \\ & \text{subject to } \theta + \nu_i \geq |y_i - \mathbf{x}'_i \boldsymbol{\beta}|, \quad i = 1, \dots, n \\ & \nu_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

Let  $\boldsymbol{\beta}_{k+1}$  denote a minimizer of Problem (3.6). This leads to an iterative optimization procedure as described in [Algorithm 1](#).

We next study the convergence properties of [Algorithm 1](#).

---

**Algorithm 1** Sequential Linear Optimization Algorithm for the LQS problem
 

---

- 1 Initialize with  $\beta_1$ , and for  $k \geq 1$  perform the following Steps 2-3 for predefined tolerance parameter “Tol”.
  - 2 Solve the linear optimization problem (3.6) and let  $(\nu_{k+1}, \theta_{k+1}, \beta_{k+1})$  denote a minimizer.
  - 3 If  $(|y_{(q)} - \mathbf{x}'_{(q)}\beta_k| - |y_{(q)} - \mathbf{x}'_{(q)}\beta_{k+1}|) \leq \text{Tol} \cdot |y_{(q)} - \mathbf{x}'_{(q)}\beta_k|$  exit; else go to Step 2.
- 

*Convergence Analysis of Algorithm 1.* In representation (3.1), we replace  $H_{q+1}(\beta)$  by its dual representation (3.3) to obtain:

$$(3.7) \quad \begin{aligned} f_q(\beta) := & \min_{\nu, \theta} & F(\nu, \theta, \beta) := & \theta(q+1) + \sum_{i=1}^n \nu_i - H_q(\beta) \\ & \text{subject to} & & \theta + \nu_i \geq |y_i - \mathbf{x}'_i \beta|, & i = 1, \dots, n \\ & & & \nu_i \geq 0, & i = 1, \dots, n. \end{aligned}$$

This leads to

$$(3.8) \quad \begin{aligned} \min_{\beta} |r_{(q)}| = \min_{\beta} f_q(\beta) = \min_{\nu, \theta, \beta} & F(\nu, \theta, \beta) \\ \text{s.t.} & \theta + \nu_i \geq |y_i - \mathbf{x}'_i \beta|, \quad i = 1, \dots, n \\ & \nu_i \geq 0, \quad i = 1, \dots, n \end{aligned}$$

The objective function  $F(\nu, \theta, \beta)$  appearing in (3.7) is the sum of a linear function in  $(\nu, \theta)$  and a concave function in  $\beta$  and the constraints are convex.

Note that the function:

$$(3.9) \quad Q((\nu, \theta, \beta); \bar{\beta}) = \theta(q+1) + \sum_{i=1}^n \nu_i - \langle \partial H_q(\bar{\beta}), \beta - \bar{\beta} \rangle - H_q(\bar{\beta}),$$

which is linear in the variables  $(\nu, \theta, \beta)$  is a linearization of  $F(\nu, \theta, \beta)$  at the point  $\bar{\beta}$ . Since  $H_q(\beta)$  is convex in  $\beta$ , the function  $Q((\nu, \theta, \beta); \bar{\beta})$  is a majorizer of  $F(\nu, \theta, \beta)$  for *any* fixed  $\bar{\beta}$  with equality holding at  $\bar{\beta} = \beta$ , i.e.,

$$Q((\nu, \theta, \beta); \bar{\beta}) \geq F(\nu, \theta, \beta), \forall \beta, \quad \text{and} \quad Q((\nu, \theta, \bar{\beta}); \bar{\beta}) = F(\nu, \theta, \bar{\beta}).$$

Observe that Problem (3.6) minimizes the function  $Q((\nu, \theta, \beta); \beta_k)$ .

It follows that for every fixed  $\bar{\beta}$ , an optimal solution of the following linear optimization problem:

$$(3.10) \quad \begin{aligned} \min_{\nu, \theta, \beta} & Q((\nu, \theta, \beta); \bar{\beta}) \\ \text{subject to} & \theta + \nu_i \geq |y_i - \mathbf{x}'_i \beta|, \quad i = 1, \dots, n \\ & \nu_i \geq 0, \quad i = 1, \dots, n, \end{aligned}$$

provides an upper bound to the minimum of the problem (3.8) and hence the global minimum of the LQS objective function. We now define the first order optimality conditions of Problem (3.7):

DEFINITION 1. A point  $(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$  satisfies the first order optimality conditions for the minimization problem (3.8) if (a)  $(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$  is feasible for problem (3.7) and (b)  $(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$  is a minimizer of the following LO problem

$$(3.11) \quad \begin{aligned} \Delta_* := \quad & \min_{\boldsymbol{\nu}, \theta, \boldsymbol{\beta}} \quad \left\langle \nabla F(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*), \begin{pmatrix} \boldsymbol{\nu} - \boldsymbol{\nu}_* \\ \theta - \theta_* \\ \boldsymbol{\beta} - \boldsymbol{\beta}_* \end{pmatrix} \right\rangle \\ & \text{subject to } \theta + \nu_i \geq |y_i - \mathbf{x}'_i \boldsymbol{\beta}|, & i = 1, \dots, n \\ & \nu_i \geq 0, & i = 1, \dots, n, \end{aligned}$$

where  $\nabla F(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$  is a subgradient of the function  $F(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$ .

REMARK 1. Note that if  $(\boldsymbol{\nu}_*, \theta_*, \boldsymbol{\beta}_*)$  satisfies the first order optimality conditions for the minimization problem (3.8), then  $\boldsymbol{\beta}_*$  satisfies the first order stationarity conditions for the LQS minimization problem (1.4).

Let us define  $\Delta_k$  as a measure of sub-optimality of the tuple  $(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k)$  from first order stationary conditions, given in Definition 1

$$(3.12) \quad \Delta_k := \left\langle \nabla F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k), \begin{pmatrix} \boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k \\ \theta_{k+1} - \theta_k \\ \boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k \end{pmatrix} \right\rangle.$$

Note that  $\Delta_k \leq 0$ . If  $\Delta_k = 0$ , then the point  $(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k)$  satisfies the first order stationary conditions. If  $\Delta_k < 0$ , then we can improve the solution further. The following theorem presents the rate at which  $\Delta_k \rightarrow 0$ .

THEOREM 3.1. (a) The sequence  $(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k)$  generated by Algorithm 1 leads to a decreasing sequence of objective values  $F(\boldsymbol{\nu}_{k+1}, \theta_{k+1}, \boldsymbol{\beta}_{k+1}) \leq F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k), k \geq 1$  that converges to a value  $F_*$ .

(b) The measure of sub-optimality  $\{\Delta_k\}_{K \geq k \geq 1}$  admits a  $O(1/K)$  convergence rate, i.e.,

$$\frac{F(\boldsymbol{\nu}_1, \theta_1, \boldsymbol{\beta}_1) - F_*}{K} \geq \min_{k=1, \dots, K} (-\Delta_k),$$

where  $F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k) \downarrow F_*$ .

(c) As  $K \rightarrow \infty$  the sequence satisfies the first order stationary conditions (1) for problem (3.8).

PROOF. Since the objective function in (3.10) is a linearization of the concave function (3.7) Algorithm 1 leads to a decreasing sequence of objective values:

$$f_q(\boldsymbol{\beta}_{k+1}) = F(\boldsymbol{\nu}_{k+1}, \theta_{k+1}, \boldsymbol{\beta}_{k+1}) \leq F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k) = f_q(\boldsymbol{\beta}_k).$$

Furthermore, the concavity of  $F(\boldsymbol{\nu}, \theta, \boldsymbol{\beta})$  gives rise to the following inequality:

$$(3.13) \quad F(\boldsymbol{\nu}_{k+1}, \theta_{k+1}, \boldsymbol{\beta}_{k+1}) - F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k) \leq \left\langle \nabla F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k), \begin{pmatrix} \boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k \\ \theta_{k+1} - \theta_k \\ \boldsymbol{\beta}_{k+1} - \boldsymbol{\beta}_k \end{pmatrix} \right\rangle.$$

Considering inequality (3.13) for  $k = 1, \dots, K$  and adding up the terms we have:

$$(3.14) \quad \sum_{k=1}^K (F(\boldsymbol{\nu}_k, \theta_k, \boldsymbol{\beta}_k) - F(\boldsymbol{\nu}_{k+1}, \theta_{k+1}, \boldsymbol{\beta}_{k+1})) \geq \sum_{k=1}^K (-\Delta_k)$$

$$(3.15) \quad \text{i.e.,} \quad F(\boldsymbol{\nu}_1, \theta_1, \boldsymbol{\beta}_1) - F(\boldsymbol{\nu}_{K+1}, \theta_{K+1}, \boldsymbol{\beta}_{K+1}) \geq K \left( \min_{k=1, \dots, K} (-\Delta_k) \right)$$

$$(3.16) \quad \text{i.e.,} \quad \frac{F(\boldsymbol{\nu}_1, \theta_1, \boldsymbol{\beta}_1) - F_*}{K} \geq \left( \min_{k=1, \dots, K} (-\Delta_k) \right).$$

Equation (3.16) provides a convergence rate for the algorithm. As  $K \rightarrow \infty$ , we see that  $\Delta_k \rightarrow 0$  — corresponding to the first order stationarity condition (3.11). This also corresponds to a local minimum of (1.4).  $\square$

**3.2. First-order Subgradient Method for the LQS Problem.** Subgradient descent methods have a long history in non-smooth convex optimization (Shor, Kiwiel and Ruszcayski, 1985; Nesterov, 2004). If computation of the subgradients turns out to be inexpensive, then subgradient based methods are quite effective in obtaining a moderate accuracy solution with relatively low computational cost. For non-convex and non-smooth functions, a subgradient need not exist, so the notion of a subgradient needs to be generalized. For non-convex, non-smooth functions having certain regularity properties (for example, Lipschitz functions) subdifferentials exist and form a natural generalization of subgradients (Clarke, 1990). Algorithms based on subdifferential information oracles (see for example, Shor, Kiwiel and Ruszcayski (1985)) are thus used as natural generalizations of subgradient methods for non-smooth, non-convex optimization problems. While general subdifferential-based methods can become quite complicated based on appropriate choices of the subdifferential and step-size sequences, we propose a simple subdifferential based method for approximately minimizing  $f_q(\boldsymbol{\beta})$  as we describe below. Recall that  $f_q(\boldsymbol{\beta})$  admits a representation as the difference of two simple convex functions of the form (3.1). It follows that  $f_q(\boldsymbol{\beta})$  is Lipschitz (Rockafellar, 1996), almost everywhere differentiable and any element belonging to the set difference

$$\partial f_q(\boldsymbol{\beta}) \in \partial H_{q+1}(\boldsymbol{\beta}) - \partial H_q(\boldsymbol{\beta}),$$

where,  $\partial H_r(\boldsymbol{\beta})$  is the set of subgradients defined in (3.4); is a *subdifferential* (Shor, Kiwiel and Ruszcayski, 1985) of  $f_q(\boldsymbol{\beta})$ .

In particular, the quantity:

$$\partial f_q(\boldsymbol{\beta}) = -\text{sgn}(y_{(q)} - \mathbf{x}'_{(q)}\boldsymbol{\beta})\mathbf{x}_{(q)}$$

is a subdifferential of the function  $f_q(\boldsymbol{\beta})$  at  $\boldsymbol{\beta}$ .

Using the definitions above, we propose a first order *subgradient descent*<sup>4</sup> method for the LQS problem as described in Algorithm 2, below.

While various step-size choices are possible, we found the following simple fixed step-size sequence to be quite useful in our experiments:

$$\alpha_k = \frac{1}{\max_{i=1, \dots, n} \|\mathbf{x}_i\|_2},$$

---

<sup>4</sup>Strictly speaking this is a subdifferential based method and not a subgradient method, but we will adhere to “subgradient descent” for easier terminology.

---

**Algorithm 2** Subgradient Descent Algorithm for the LQS problem
 

---

1. Initialize  $\beta_1$ , for  $\text{MaxIter} \geq k \geq 1$  do the following:
  2.  $\beta_{k+1} = \beta_k - \alpha_k \partial f_q(\beta_k)$  where  $\alpha_k$  is a step-size.
  3. Return  $\min_{1 \leq k \leq \text{MaxIter}} f_q(\beta_k)$  and  $\beta_{k^*}$  at which the minimum is attained, where  $k^* = \arg \min_{1 \leq k \leq \text{MaxIter}} f_q(\beta_k)$ .
- 

the above quantity  $\max_{i=1, \dots, n} \|\mathbf{x}_i\|_2$  may be interpreted as an upper bound to the subdifferentials of  $f_q(\beta)$ . Similar constant step-size based rules are often used in subgradient descent methods for convex optimization.

3.3. *A Hybrid Algorithm.* Let  $\hat{\beta}_{\text{GD}}$  denote the estimate produced by Algorithm 2. Since Algorithm 2 runs with a fixed step-size, the estimate  $\hat{\beta}_{\text{GD}}$  need not be a local minimum of the LQS problem. Algorithm 1, on the other hand, delivers an estimate  $\hat{\beta}_{\text{LO}}$ , say, which is a *local* minimum of the LQS objective function. We found that if  $\hat{\beta}_{\text{GD}}$  obtained from the subgradient method is used as a warm-start for the sequential linear optimization algorithm, the estimator obtained improves upon  $\hat{\beta}_{\text{GD}}$  in terms of the LQS objective value. This leads to the proposal of a hybrid version of Algorithm 1 and Algorithm 2, as presented in Algorithm 3 below.

---

**Algorithm 3** A Hybrid Algorithm for the LQS problem
 

---

1. Run Algorithm 2 initialized with  $\beta_1$  for  $\text{MaxIter}$  iterations. Let  $\hat{\beta}_{\text{GD}}$  be the solution.
  2. Run Algorithm 1 with  $\hat{\beta}_{\text{GD}}$  as the initial solution and Tolerance parameter “Tol” to obtain  $\hat{\beta}_{\text{LO}}$ .
  3. Return  $\hat{\beta}_{\text{LO}}$  as the solution to Algorithm 3.
- 

3.4. *Initialization Strategies for the Algorithms.* Both Algorithm 1 and Algorithm 2 are sensitive to initializations  $\beta_1$ . We run each algorithm for a prescribed number of runs “RUNS” (say), and consider the solution that gives the best objective value among them. For the initializations we found two strategies to be quite useful.

*Initialization around LAD solutions.* One method is based on the LAD solution, i.e.,  $\hat{\beta}^{(\text{LAD})}$  and random initializations around  $\hat{\beta}^{(\text{LAD})}$  given by  $\left[ \hat{\beta}_i^{(\text{LAD})} - \eta |\hat{\beta}_i^{(\text{LAD})}|, \hat{\beta}_i^{(\text{LAD})} + \eta |\hat{\beta}_i^{(\text{LAD})}| \right]$ , for  $i = 1, \dots, p$ , where  $\eta$  is a predefined number say  $\eta \in \{2, 4\}$ . This initialization strategy leads to  $\beta^1$ , which we denote by the “LAD” initialization.

*Initialization around Chebyshev fits.* Another initialization strategy is inspired by a geometric characterization of the LQS solution (see Stromberg (1993) and also Section 4). Consider a sub-sample  $\mathcal{J} \subset \{1, \dots, n\}$  of size of size  $(p + 1)$  and the associated  $\ell_\infty$  regression fit (also known as the Chebyshev fit) on the sub-sample  $(y_i, \mathbf{x}_i), i \in \mathcal{J}$  given by

$$\hat{\beta}_{\mathcal{J}} \in \arg \min_{\beta} \left( \max_{i \in \mathcal{J}} |y_i - \mathbf{x}_i' \beta| \right).$$

Consider a number of random subsamples  $\mathcal{J}$  and the associated coefficient-vector  $\hat{\beta}_{\mathcal{J}}$  for every  $\mathcal{J}$ . The estimate  $\hat{\beta}_{\mathcal{J}^*}$  that produces the minimum value of the LQS objective function is taken as  $\beta_1$ . We denote  $\beta_1$  chosen in this fashion as the best Chebyshev fit or “Cheb” in short.

Algorithm 3, in our experience was found to be less sensitive to initializations. Experiments demonstrating the different strategies described above are discussed in Section 5.

**4. Properties of the LQS Solutions for Arbitrary Datasets.** In this section, we prove that key properties of optimal LQS solutions hold without assuming that the data  $(\mathbf{y}, \mathbf{X})$  are in general position as it is done in the literature to date (Rousseeuw, 1984; Rousseeuw and Leroy, 1987; Stromberg, 1993). For this purpose we utilize the MIO characterization of the LQS problem. Specifically,

1. We show in Theorem 4.1 that an optimal solution to the LQS problem (and in particular the LMS problem) always exists, for *any*  $(\mathbf{y}, \mathbf{X})$  and  $q$ . The theorem also shows that an optimal LQS solution is given by the  $\ell_\infty$  or Chebyshev regression fit to a subsample of size  $q$  from the sample  $(y_i, \mathbf{x}_i), i = 1, \dots, n$ , thereby generalizing the results of Stromberg (1993), which require  $(\mathbf{y}, \mathbf{X})$  to be in general position.
2. We show in Theorem 4.2 that the absolute values of some of the residuals are equal to the optimal solution value of the LQS problem, without assuming that the data is in general position.
3. We show in Theorem 4.3 a new result that the breakdown point of the optimal value of the LQS objective is  $(n - q + 1)/n$  without assuming that the data is in general position. For the LMS problem  $q = n - \lfloor n/2 \rfloor$ , which leads to the sample breakdown point of LQS optimal objective value of  $(\lfloor n/2 \rfloor + 1)/n$ , independent of the number of covariates  $p$ . In contrast, LMS solutions have a sample breakdown point of  $(\lfloor n/2 \rfloor - p + 2)/n$ .

**THEOREM 4.1.** *The LQS problem is equivalent to the following problem*

$$(4.1) \quad \min_{\boldsymbol{\beta}} |r_{(q)}| = \min_{\mathcal{I} \in \Omega_q} \left( \min_{\boldsymbol{\beta}} \|\mathbf{y}_{\mathcal{I}} - \mathbf{X}_{\mathcal{I}}\boldsymbol{\beta}\|_{\infty} \right),$$

where,  $\Omega_q := \{\mathcal{I} : \mathcal{I} \subset \{1, \dots, n\}, |\mathcal{I}| = q\}$  and  $(\mathbf{y}_{\mathcal{I}}, \mathbf{X}_{\mathcal{I}})$  denotes the subsample  $(y_i, \mathbf{x}_i), i \in \mathcal{I}$ .

**PROOF.** Consider the MIO formulation (2.11) for the LQS problem. Let us take a vector of binary variables  $\bar{z}_i \in \{0, 1\}, i = 1, \dots, n$  with  $\sum_i \bar{z}_i = q$ , feasible for problem (2.11). This vector  $\bar{\mathbf{z}} := (\bar{z}_1, \dots, \bar{z}_n)$  gives rise to a subset  $\mathcal{I} \in \Omega_q$  given by:

$$\mathcal{I} = \{i \mid \bar{z}_i = 1, i \in \{1, \dots, n\}\}.$$

Corresponding to this subset  $\mathcal{I}$  consider the subsample  $(y_{\mathcal{I}}, \mathbf{X}_{\mathcal{I}})$  and the associated optimization problem:

$$(4.2) \quad T_{\mathcal{I}} = \min_{\boldsymbol{\beta}} \|\mathbf{y}_{\mathcal{I}} - \mathbf{X}_{\mathcal{I}}\boldsymbol{\beta}\|_{\infty},$$

and let  $\boldsymbol{\beta}_{\mathcal{I}}$  be a minimizer of (4.2). Observe that  $\bar{\mathbf{z}}, \boldsymbol{\beta}_{\mathcal{I}}$  and  $\bar{r}_i = y_i - \mathbf{x}_i' \boldsymbol{\beta}_{\mathcal{I}}, i = 1, \dots, n$  is feasible for problem (2.11). Furthermore, it is easy to see that, if  $\mathbf{z}$  is taken to be equal to  $\bar{\mathbf{z}}$ , then the minimum value of problem (2.11) with respect to the variables  $\boldsymbol{\beta}$  and  $r_i^+, r_i^-, \mu_i, \bar{\mu}_i$  for  $i = 1, \dots, n$  is given by  $|\bar{r}_{(q)}| = T_{\mathcal{I}}$ . Since every choice of  $\mathbf{z} \in \{0, 1\}^n$  with  $\sum_i z_i = q$  corresponds to a subset  $\mathcal{I} \in \Omega_q$ , it follows that the minimum value of problem (2.11) is given by the minimum value of  $T_{\mathcal{I}}$  as  $\mathcal{I}$  varies over  $\Omega_q$ .

Note that the minimum in problem (4.1) is attained since it is a minimum over finitely many subsets  $\mathcal{I} \in \Omega_q$ . This shows that an optimal solution to the LQS problem always exists, without any assumption on the geometry or orientation of the sample points  $(\mathbf{y}, \mathbf{X})$ . This completes the proof of the equivalence (4.1).  $\square$

COROLLARY 1. *Theorem 4.1 shows that an optimal LQS solution for any sample  $(\mathbf{y}, \mathbf{X})$  is given by the Chebyshev or  $\ell_\infty$  regression fit to a subsample of size  $q$  from the  $n$  sample points. In particular, for every optimal LQS solution there is a  $\mathcal{I}_* \in \Omega_q$  such that*

$$(4.3) \quad \widehat{\boldsymbol{\beta}}^{(LQS)} \in \arg \min_{\boldsymbol{\beta}} \|\mathbf{y}_{\mathcal{I}_*} - \mathbf{X}_{\mathcal{I}_*} \boldsymbol{\beta}\|_\infty.$$

We next show that, at an optimal solution of the LQS problem, some of the absolute values of the residuals are all equal to the minimum objective value of the LQS problem, generalizing earlier work by Stromberg (1993). Note that Problem (4.2) can be written as the following linear optimization problem:

$$(4.4) \quad \underset{t, \boldsymbol{\beta}}{\text{minimize}} \quad t \quad \text{subject to} \quad -t \leq y_i - \mathbf{x}'_i \boldsymbol{\beta} \leq t, i \in \mathcal{I}_*.$$

The Karush Kuhn Tucker (KKT) (Boyd and Vandenberghe, 2004) optimality conditions of the problem (4.4) are given by:

$$(4.5) \quad \begin{aligned} \sum_{i \in \mathcal{I}_*} (\nu_i^- + \nu_i^+) &= 1 \\ \sum_{i \in \mathcal{I}_*} (\nu_i^- - \nu_i^+) \mathbf{x}_i &= \mathbf{0} \\ \nu_i^+ (y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}} - t^*) &= 0, \quad \forall i \in \mathcal{I}_* \\ \nu_i^- (y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}} + t^*) &= 0, \quad \forall i \in \mathcal{I}_* \\ \nu_i^+, \nu_i^- &\geq 0, \quad \forall i \in \mathcal{I}_*, \end{aligned}$$

where  $\widehat{\boldsymbol{\beta}}, t^*$  are optimal solutions<sup>5</sup> to (4.4).

Let us denote

$$(4.6) \quad \mathcal{I}^+ := \{i \mid i \in \mathcal{I}_*, \nu_i^+ > 0, \nu_i^- > 0\},$$

clearly, on this set of indices  $|y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}}| = t^*$ . This gives the following bound

$$|\mathcal{I}^+| \leq \left| \left\{ i \in \mathcal{I}_* : |y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}}| = t^* \right\} \right|.$$

It follows from (4.5) that  $|\mathcal{I}^+| > \text{rank}([\mathbf{x}_i, i \in \mathcal{I}^+])$ . We thus have:

$$\left| \left\{ i \in \mathcal{I}_* : |y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}}| = t^* \right\} \right| \geq |\mathcal{I}^+| > \text{rank}([\mathbf{x}_i, i \in \mathcal{I}^+]).$$

In particular, if the  $\mathbf{x}_i$ 's come from a continuous distribution, then with probability one:

$$\text{rank}([\mathbf{x}_i, i \in \mathcal{I}^+]) = p \quad \text{and} \quad \left| \left\{ i \in \mathcal{I}_* : |y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}}| = t^* \right\} \right| \geq (p + 1).$$

This leads to the following theorem.

THEOREM 4.2. *Let  $\mathcal{I}_* \in \Omega_q$  denote the subset of size  $q$  which corresponds to the set of optimal LQS solutions (see Corollary 1). Consider the KKT optimality conditions of the Chebyshev fit to this sub-sample  $(\mathbf{y}_{\mathcal{I}_*}, \mathbf{X}_{\mathcal{I}_*})$  as given by (4.5). Then:*

$$\left| \left\{ i \in \mathcal{I}_* : |y_i - \mathbf{x}'_i \widehat{\boldsymbol{\beta}}| = t^* \right\} \right| \geq |\mathcal{I}^+| > \text{rank}([\mathbf{x}_i, i \in \mathcal{I}^+]),$$

where  $\widehat{\boldsymbol{\beta}}, \mathcal{I}^+$  are as defined in (4.6).

---

<sup>5</sup>We use the shorthand  $\widehat{\boldsymbol{\beta}}$  in place of  $\widehat{\boldsymbol{\beta}}^{(LQS)}$ .

4.1. *Breakdown Point and Stability of Solutions.* In this section, we revisit the notion of a breakdown point of estimators and derive results about the breakdown point of the LQS optimal objective function value without the assumption that the data is in general position. Suppose the original sample is  $(\mathbf{y}, \mathbf{X})$  and  $m$  of the sample points have been replaced arbitrarily—let  $(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})$  denote the perturbed sample. Let

$$(4.7) \quad \alpha(m; \Theta; (\mathbf{y}, \mathbf{X})) = \sup_{(\Delta_{\mathbf{y}}, \Delta_{\mathbf{X}})} \|\Theta(\mathbf{y}, \mathbf{X}) - \Theta(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})\|,$$

denote the maximal change in the estimator under this perturbation, where  $\|\cdot\|$  denotes the standard Euclidean norm. The finite sample breakdown point of the estimator  $\Theta$  is defined as follows:

$$(4.8) \quad \eta(\Theta; (\mathbf{y}, \mathbf{X})) := \min_m \left\{ \frac{m}{n} \mid \alpha(m; \Theta; (\mathbf{y}, \mathbf{X})) = \infty \right\}.$$

We will derive the breakdown point of the minimum value of the LQS objective function, i.e.,  $|r_{(q)}| = |y_{(q)} - \mathbf{x}'_{(q)} \widehat{\boldsymbol{\beta}}^{(LQS)}|$ , as defined in (3.1).

**THEOREM 4.3.** *Let  $\widehat{\boldsymbol{\beta}}^{(LQS)}$  denote an optimal solution and  $\Theta := \Theta(\mathbf{y}, \mathbf{X})$  denote the optimum objective value to the LQS problem for a given dataset  $(\mathbf{y}, \mathbf{X})$ , where the  $(y_i, \mathbf{x}_i)$ 's are not necessarily in general position. Then, the finite sample breakdown point of  $\Theta$  is  $(n - q + 1)/n$ .*

**PROOF.** We will first show that the breakdown point of  $\Theta$  is strictly greater than  $(n - q)/n$ . Suppose we have a corrupted sample  $(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})$ , with  $m = n - q$  replacements in the original sample. Consider the equivalent LQS formulation (4.1) and let  $\mathcal{I}_0$  denote the unchanged sample indices. Consider the inner convex optimization problem appearing in (4.1), corresponding to the index set  $\mathcal{I}_0$ :

$$(4.9) \quad T_{\mathcal{I}_0}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}}) = \min_{\boldsymbol{\beta}} \|\mathbf{y}_{\mathcal{I}_0} - \mathbf{X}_{\mathcal{I}_0} \boldsymbol{\beta}\|_{\infty},$$

with  $\boldsymbol{\beta}_{\mathcal{I}_0}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})$  denoting a minimizer of the convex optimization problem (4.9).

Clearly, both a minimizer and the minimum objective value are finite and neither depends upon  $(\Delta_{\mathbf{y}}, \Delta_{\mathbf{X}})$ . Suppose

$$T_{\mathcal{I}^*}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}}) = \min_{\mathcal{I} \in \Omega_q} T_{\mathcal{I}}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})$$

denotes the minimum value of the LQS objective function corresponding to the perturbed sample, for some  $\mathcal{I}^* \in \Omega_q$ , then it follows that:  $T_{\mathcal{I}^*}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}}) \leq T_{\mathcal{I}_0}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}})$ —which clearly implies that the quantity  $\|T_{\mathcal{I}^*}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}}) - \Theta\|$  is bounded above and the bound does not depend upon  $(\Delta_{\mathbf{y}}, \Delta_{\mathbf{X}})$ . This shows that the breakdown point of  $\Theta$  is strictly larger than  $\frac{(n-q)}{n}$ .

We will now show that the breakdown point of the estimator is less than or equal to  $(n - q + 1)/n$ . If the number of replacements is given by  $m = n - q + 1$ , then it is easy to see that every  $\mathcal{I} \in \Omega_q$  includes a sample  $i_0$  (say) from the replaced sample units. If  $(\delta_{y_{i_0}}, \delta'_{\mathbf{x}_{i_0}})$  denotes the perturbation corresponding to the  $i_0$ th sample, then, it is easy to see that:

$$T_{\mathcal{I}}(\mathbf{y} + \Delta_{\mathbf{y}}, \mathbf{X} + \Delta_{\mathbf{X}}) \geq |(y_{i_0} - \mathbf{x}_{i_0} \boldsymbol{\beta}_{\mathcal{I}}) + (\delta_{y_{i_0}} - \delta'_{\mathbf{x}_{i_0}} \boldsymbol{\beta}_{\mathcal{I}})|.$$

It is possible to choose  $\delta_{y_{i_0}}$  such that the r.h.s. of the above inequality becomes arbitrarily large. Thus, the finite-sample breakdown point of the estimator  $\Theta$  is  $\frac{(n-q+1)}{n}$ .  $\square$

For the LMS problem  $q = n - \lfloor n/2 \rfloor$ , which leads to the sample breakdown point of  $\Theta$  of  $(\lfloor n/2 \rfloor + 1)/n$ , independent of the number of covariates  $p$ . In contrast, LMS solutions have a sample breakdown point of  $(\lfloor n/2 \rfloor - p + 2)/n$ . In other words, the optimal solution value is more robust than optimal solutions to the LMS problem.

**5. Computational Experiments.** In this section, we perform computational experiments demonstrating the effectiveness of our algorithms in terms of quality of solutions obtained, scalability and speed.

All computations were done in MATLAB version 7.12.0.635 (R2011a) on a 64-bit linux machine, with 8 cores and 32 GB RAM. For the MIO formulations we used GUROBI ([Gurobi Optimization, 2013](#)) via its MATLAB interface.

We consider a series of examples including synthetic and real-world datasets showing that our proposed methodology consistently finds high quality solutions of problems of sizes up to  $n = 10,000$  and  $p = 20$ . We observed that global optimum solutions are obtained usually within a few minutes (or even faster) in all these examples, but it takes longer to deliver a certificate of global optimality. Our continuous optimization based methods enhance the performance of the MIO formulation, the margin of improvement becomes more significant with increasing problem sizes. In all the examples, there is an appealing common theme — if the MIO algorithm is terminated early, the procedure provides a bound on its sub-optimality.

In Section 5.1 we describe the synthetic datasets used in our experiments. Section 5.2 studies the performances of Algorithms 1, 2 and 3 on synthetic datasets. Section 5.3 presents comparisons of Algorithms 1, 2 and 3 as well as the MIO algorithm with state of the art algorithms for the LQS. In section 5.4 we illustrate the performance of our algorithms on real-world data sets. Section 5.5 discusses the evolution of lower bounds and global convergence certificates for the problem. Section 5.6 describes scalability considerations for larger problems.

**5.1. Synthetic Examples.** We considered a set of synthetic examples, following [Rousseeuw and Driessen \(2006\)](#). We generated the model matrix  $\mathbf{X}_{n \times p}$  with iid Gaussian entries  $N(0, 100)$  and took  $\boldsymbol{\beta} \in \mathbb{R}^p$  to be a vector of all ones. Subsequently, the response is generated as  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ , where  $\epsilon_i \sim N(0, 10), i = 1, \dots, n$ . Once  $(\mathbf{y}, \mathbf{X})$  have been generated, we corrupt a certain proportion  $\pi$  of the sample in two different ways:

- (A)  $\lfloor \pi n \rfloor$  of the samples are chosen at random and the first coordinate of the data matrix  $\mathbf{X}$ , i.e.,  $x_{1j}$ 's are replaced by  $x_{1j} \leftarrow x_{1j} + 1000$ .
- (B)  $\lfloor \pi n \rfloor$  of the samples are chosen at random out of which the covariates of half of the points are changed as in Item (A); for the remaining half of the points the responses are corrupted as  $y_j \leftarrow y_j + 1000$ . In this set-up outliers are added in *both* the covariate and response spaces.

Ex- We considered seven different examples for different values of  $(n, p, \pi)$ :

**Moderate-Scale:** We consider four moderate-scale examples Ex-1—Ex-4:

- Ex-1: Data is generated as per (B) with  $(n, p, \pi) = (201, 5, 0.4)$ .
- Ex-2: Data is generated as per (B) with  $(n, p, \pi) = (201, 10, 0.5)$ .
- Ex-3: Data is generated as per (A) with  $(n, p, \pi) = (501, 5, 0.4)$ .
- Ex-4: Data is generated as per (A) with  $(n, p, \pi) = (501, 10, 0.4)$ .

**Large-Scale:** We consider three large scale examples, Ex-5—Ex-7:

- Ex-5: Data is generated as per (B) with  $(n, p, \pi) = (2001, 10, 0.4)$ .

Ex-6: Data is generated as per (B) with  $(n, p, \pi) = (5001, 10, 0.4)$ .

Ex-7: Data is generated as per (B) with  $(n, p, \pi) = (10001, 20, 0.4)$ .

5.2. *A Deeper Understanding of Algorithms 1, 2 and 3.* For each of the synthetic examples Ex-1—Ex-4, we compared the performances of the different continuous optimization based algorithms proposed in this paper—Algorithm 1, Algorithm 2 and Algorithm 3. For each of the Algorithms 1, 2 we considered two different initializations, following the strategy described Section 3.4:

- (LAD) This is the initialization from the LAD solution, with  $\eta = 2$  and number of random initializations taken to be 100. This is denoted in Table 1 by the moniker “LAD”.
- (Cheb) This is the initialization from the Chebyshev fit. For every initialization, forty different sub-samples were taken to estimate  $\beta_1$ , 100 different initializations were considered. This method is denoted by the moniker “Cheb” in Table 1.

Algorithm 1, initialized at the “LAD” method (described above) is denoted by Algorithm 1 (LAD), the same notation carries over to the other remaining combinations of Algorithms 1 and 2 with initializations “LAD” and “Cheb”. Each of the methods Algorithm 2 (LAD) and Algorithm 2 (Cheb), leads to an initialization for Algorithm 3—denoted by Algorithm 3 (LAD) and Algorithm 3 (Cheb), respectively.

In all the examples, we set the Maxiter counter for Algorithm 2 at 500 and took the step-size sequence as described in Section 3.2. The tolerance criterion “Tol” used in Algorithm 1 (and consequently Algorithm 3), was set to  $10^{-4}$ .

Results comparing these methods are summarized in Table 1. To compare the different algorithms in terms of the quality of solutions obtained, we do the following. For every instance, we run all the algorithms and obtain the best solution among them, say,  $f_*$ . If  $f_{\text{alg}}$  denotes the value of the LQS objective function for Algorithm “alg”, then we define the relative accuracy of the solution obtained by “alg” as:

$$(5.1) \quad \text{Relative Accuracy} = (f_{\text{alg}} - f_*)/f_* \times 100.$$

To obtain the entries in Table 1 the relative accuracy is computed for every algorithm (six in all: Algorithm 1—3, two types for each “LAD” and “Cheb”) for every random problem instance corresponding to a particular example type; and the results are averaged (over 20 runs). The times reported for Algorithm 1 (LAD) and Algorithm 1 (Cheb) includes the times taken to perform the LAD and Chebyshev fits, respectively. The same thing applies to Algorithm 2 (LAD) and Algorithm 2 (Cheb). For Algorithm 3 (Cheb) (respectively, Algorithm 3 (LAD)) the time taken equals the time taken by Algorithm 2 (Cheb) (respectively, Algorithm 2 (LAD)) and the time taken to perform the Chebyshev (respectively, LAD) fits.

In Table 1, we see that Algorithm 2 (LAD) converges quite quickly in all the examples. The quality of the solution, however, depends upon the choice of  $p$ —for  $p = 10$  the algorithm converges to a lower quality solution when compared to  $p = 5$ . The time till convergence for Algorithm 2 is less sensitive to the problem dimensions—this is in contrast to the other Algorithms, where computation times show a monotone trend depending upon the sizes of  $(n, p)$ . Algorithm 2 (Cheb) takes more time than Algorithm 2 (LAD), since it spends a considerable amount of time in performing multiple Chebyshev fits (to obtain a good initialization). Algorithm 1 (LAD) seems to be sensitive to the type of initialization used; Algorithm 1 (Cheb) is more stable and it appears that the multiple Chebyshev initialization guides Algorithm 1 (Cheb) to higher quality solutions. Algorithm 3 (both variants) seem to be the clear winner among the various algorithms—this does not come as a

surprise since, intuitively it aims at combining the *best features* of its constituent algorithms. Based on computation times, Algorithm 3 (LAD) outperforms Algorithm 3 (Cheb), since it avoids the computational overhead of computing several Chebyshev fits.

Example ( $n, p, \pi$ )	q	Algorithm Used					
		Algorithm 1		Algorithm 2		Algorithm 3	
		(LAD)	(Cheb)	(LAD)	(Cheb)	(LAD)	(Cheb)
Ex-1 (201,5, 0.4)	Accuracy	49.399 (2.43)	0.0 (0.0)	0.233 (0.03)	0.240 (0.02)	0.0 (0.0)	0.0 (0.0)
q=121	Time (s)	24.05	83.44	3.29	83.06	36.13	118.43
Ex-2 (201,10, 0.5)	Accuracy	43.705 (2.39)	5.236 (1.73)	1.438 (0.07)	1.481 (0.10)	0.0 (0.0)	0.0 (0.0)
q=101	Time (s)	54.39	133.79	3.22	73.14	51.89	125.55
Ex-3 (501,5,0.4)	Accuracy	2.897 (0.77)	0.0 (0.0)	0.249 (0.05)	0.274 (0.06)	0.0 (0.0)	0.0 (0.0)
q=301	Time (s)	83.01	158.41	3.75	62.36	120.90	179.34
Ex-4 (501,10, 0.4)	Accuracy	8.353 (2.22)	11.926 (2.31)	1.158 (0.06)	1.083 (0.06)	0.0	0.0
q=301	Time (s)	192.02	240.99	3.76	71.45	155.36	225.09

TABLE 1

Table showing performances of different continuous optimization based methods proposed in this paper for Examples Ex-1–Ex-4. For every example, the top row “Accuracy” is Relative Accuracy (see (5.1)) and the numbers inside parenthesis denotes standard errors (across the random runs); the lower row denotes the time taken (in cpu seconds). Results are averaged over 20 different random instances of the problem. Algorithm 3 seems to be the clear winner among the different examples, in terms of the quality of solutions obtained. Among all the algorithms considered, Algorithm 3 seems to be least sensitive to initializations.

5.3. *Comparisons : Quality of the Solutions Obtained.* In this section, we shift our focus from studying the detailed dynamics of Algorithms 2–3; and compare the performances of Algorithm 3 (which seems to be the best among the algorithms considered in the paper), the MIO formulation (2.11) and state-of-the art implementations of the LQS problem as implemented in the popular R-package MASS (available from CRAN). For the MIO formulation (2.11), we considered two variations: MIO formulation (2.11)(cold-start), where the MIO algorithm is not provided with any advanced warm-start) and MIO formulation (2.11)(warm-start), where the MIO algorithm is provided with an advanced warm-start obtained by Algorithm 3.

The focus here is on comparing the quality of upper bounds to the LQS problem. We consider the same datasets used in Section 5.2 for our experiments. The results are shown in Table 2. We see that MIO formulation (2.11)(warm-start) is the clear winner among all the examples, Algorithm 3 comes a close second. MIO formulation (2.11) (cold-start) does seem to benefit significantly from advanced warm-starts as provided by Algorithm 3. The state-of-the art algorithm LQS delivers a solution very quickly, but the solutions obtained are quite far from the global minimum.

5.4. *Performance on Some Real-world Datasets.* We considered a few real-world datasets popularly used in the context of robust statistical estimation, as available from the R package robustbase (Rousseeuw et al., 2013; Todorov and Filzmoser, 2009). We used the “Alcohol” dataset (available from the same package), which is aimed at studying the solubility of alcohols in water to understand alcohol transport in living organisms. This dataset contains physicochemical characteristics of  $n = 44$  aliphatic alcohols and measurements on seven numeric variables: SAG solvent accessible surface-bounded molecular volume ( $x_1$ ), logarithm of of the octanol-water partitions coefficient ( $x_2$ ), polarizability ( $x_3$ ), molar refractivity ( $x_4$ ), mass ( $x_5$ ), volume ( $x_6$ ) and the response ( $y$ ) is taken to be the logarithm of the solubility. We consider two cases from the Alcohol dataset—the first one

Example ( $n, p, \pi$ ) q		Algorithm Used			
		LQS (MASS)	Algorithm-3	MIO formulation (2.11) (cold-start) (warm-start)	
Ex-1 (201,5, 0.4) q=121	Accuracy	24.163 (1.31)	0.0 (0.0)	60.880 (5.60)	0.0 (0.0)
	Time (s)	0.02	36.13	71.46	35.32
Ex-2 (201,10, 0.5) q=101	Accuracy	105.387 (5.26)	0.263 (0.26)	56.0141 (3.99)	0.0 (0.0)
	Time (s)	0.05	51.89	193.00	141.10
Ex-3 (501,5,0.4) q=301	Accuracy	9.677 (0.99)	0.618 (0.27)	11.325 (1.97)	0.127 (0.11)
	Time (s)	0.05	120.90	280.66	159.76
Ex-4 (501,5,0.4) q=301	Accuracy	29.756 (1.99)	0.341 (0.33)	27.239 (2.66)	0.0 (0.0)
	Time (s)	0.08	155.36	330.88	175.52

TABLE 2

Table showing performances of various Algorithms for the LQS problem for different moderate-scale examples as described in the text. For each example, “Accuracy” is Relative Accuracy (see (5.1)) the numbers within brackets denote the standard errors; the lower row denotes the averaged cpu time (in secs) taken for the Algorithm. All results are averaged over 20 random examples.

has  $n = 44, p = 5$  where the five covariates were  $x_1, x_2, x_4, x_5, x_6$ ; the second example has all the six covariates and an intercept term, which leads to  $p = 7$ . We used the MIO formulation (2.11) (cold-start) for both the cases. The evolution of the MIO (with upper and lower bounds) for the two cases are shown in Figure 1. As expected, the time taken for the algorithm to converge is larger for  $p = 7$  than for  $p = 5$ .

We considered a second dataset created by Hawkins, Bradu and Kass (1984) and available from the R-package `robustbase`. The dataset consists of 75 observations in four dimensions (one response and three explanatory variables), i.e.,  $n = 75, p = 3$ . We computed the LQS estimate for this example for  $q \in \{60, 45\}$ . We used both the MIO formulation (2.11) (cold-start) and MIO formulation (2.11) (warm-start) and observed that the latter showed superior convergence speed to global optimality (see Figure 2). As expected, the time taken for convergence was found to increase with decreasing  $q$ -values. The results are shown in Figure 2.

5.5. *Certificate of Lower Bounds and Global Optimality.* The MIO formulation (2.11) for the LQS problem converges to the global solution. With the aid of advanced MIO warm-starts as provided by Algorithm 3 the MIO obtains a very high quality solution very quickly—in most of the examples the solution thus obtained, indeed turns out to be the global minimum. However, the certificate of global optimality comes later as the lower bounds of the problem “evolve” slowly—see for example Figures 1 and 2. We will now describe a regularized version of the MIO formulation, which we found to be quite useful in speeding up the convergence of the MIO algorithm without any loss in the accuracy of the solution. The LQS problem formulation does not contain any explicit regularization on  $\beta$ , it is rather implicit (since  $\hat{\beta}^{(\text{LQS})}$  will be bounded). We thus consider the following modified version of the LQS problem (1.4)

$$(5.2) \quad \underset{\beta}{\text{minimize}} \quad |r_{(q)}| \quad \text{subject to} \quad \|\beta - \beta_0\|_\infty \leq M$$

for some predefined  $\beta_0$  and  $M \geq 0$ . If  $\hat{\beta}_M$  solves problem (5.2), then it is the global minimum of the LQS problem in the  $\ell_\infty$ -ball  $\{\beta : -M\mathbf{1} \leq \beta - \beta_0 \leq M\mathbf{1}\}$ . In particular, if  $\hat{\beta}^{\text{LQS}}$  is the solution to problem (1.4), then by choosing  $\beta_0 = \mathbf{0}$  and  $M \geq \|\hat{\beta}^{\text{LQS}}\|_\infty$  in (5.2); both problems (1.4) and (5.2)

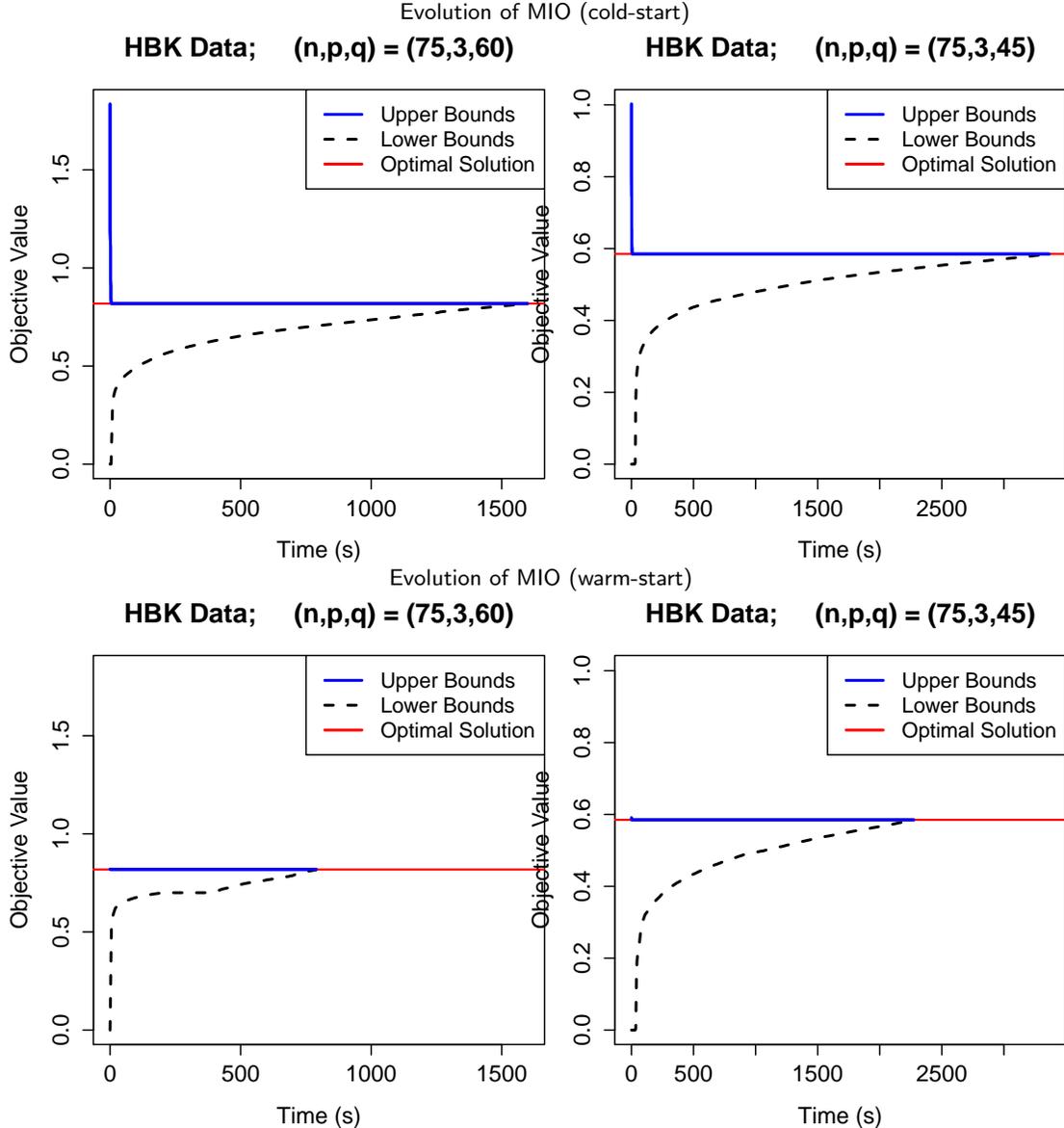


FIG 2. Figure showing the evolution of the MIO formulation (2.11) for the HBK dataset with different values of  $q$  and different warm-starts. [Top row] MIO formulation warm-started with the least squares solution, for  $q = 60$  (left panel) and  $q = 45$  (right panel). [Bottom row] MIO formulation warm-started with Algorithm 3 for  $q = 60$  (left panel) and  $q = 45$  (right panel).

will have the same solution. The MIO formulation of problem (5.2) is a very simple modification of (2.11) with additional box-constraints on  $\beta$  of the form  $\{\beta : -M\mathbf{1} \leq \beta - \beta_0 \leq M\mathbf{1}\}$ . Our empirical investigation suggests that the MIO formulation (2.11) in presence of box-constraints<sup>6</sup> produces tighter lower bounds than the unconstrained MIO formulation (2.11), for a given time limit. As an illustration of formulation (5.2), see Figure 3, where we use the MIO formulation (2.11) with box constraints. We consider two cases corresponding to  $M \in \{3, 40\}$ ; in both the cases we

<sup>6</sup>Of course, a very large value of  $M$  will render the box-constraints to be ineffective.

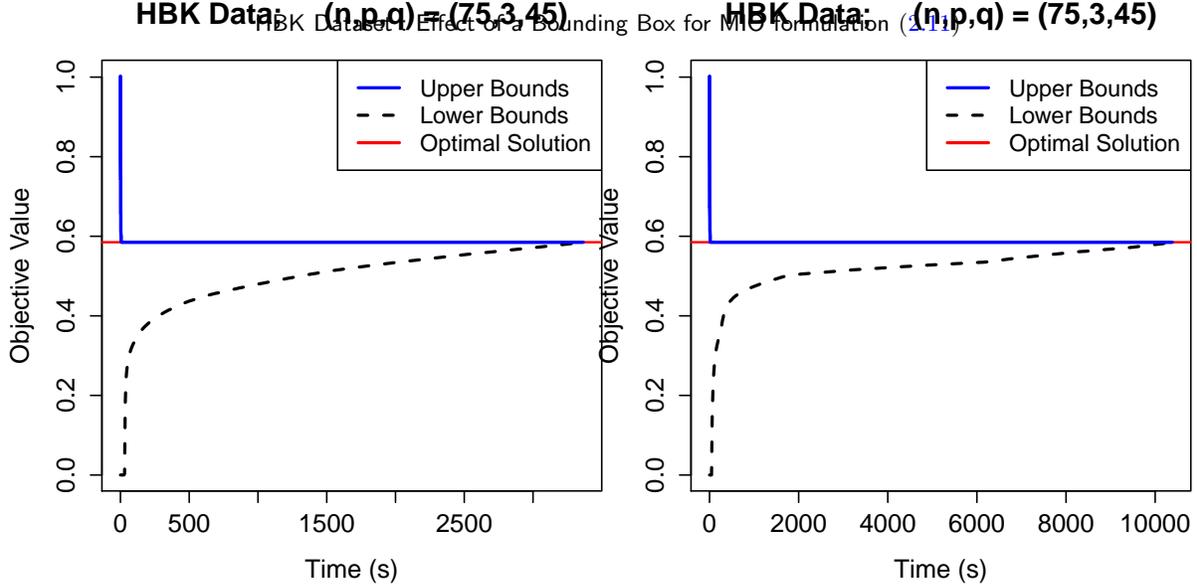


FIG 3. Figure showing the effect of the bounding box for the evolution of the MIO formulation (2.11) for the HBK dataset, with  $(n, p, q) = (75, 3, 45)$ . The left panel considers a bounding box of diameter 6 and the right panel considers a bounding box of diameter 80 centered around the least squares solution

took  $\beta_0 = \hat{\beta}^{(LS)} = (0.08, -0.36, 0.43)$ . Both these boxes (which are in fact, quite large, given that  $\|\hat{\beta}^{(LS)}\|_\infty = 0.43$ ) contains the (unconstrained) global solution for the problem. As the figure shows, the evolution of the lower bounds of the MIO algorithm towards the global optimum depends upon the radius of the box.

We argue that formulation (5.2) is a more desirable formulation—the constraint may behave as a regularizer to shrink coefficients or if one seeks an unconstrained LQS solution, there are effective ways to choose to  $\beta_0$  and  $M$ . For example, if  $\beta_0$  denotes the solution obtained by Algorithm 3, then for  $M = \eta \|\beta_0\|_\infty$ , for  $\eta \in [1, 2]$  (say), the solution to (5.2) corresponds to a global optimum of the LQS problem inside a box of diameter  $2M$  centered at  $\beta_0$ . For moderate sized problems with  $n \in \{201, 501\}$  we found this strategy to be useful in certifying global optimality within a reasonable amount of time. Figure 4 shows some examples.

**5.6. Scalability to Large Problems.** For large scale problems with  $n \geq 5000$  with  $p \geq 10$ , we found that Algorithm 1 becomes computationally demanding due to the associated LO problems (3.6) appearing in Step-2 of Algorithm 1. On the other hand, Algorithm 2 remains computationally inexpensive. So for larger problems, we propose using a modification of Algorithm 1—we run Algorithm 2 for several random initializations around the  $\hat{\beta}^{(LAD)}$  solution and find the best solution among them. The regression coefficient thus obtained is used as an initialization for Algorithm 1—we call this Algorithm 3 (large-scale). Note that in Algorithm 3, we do *both* Step 1 and Step 2 for every initialization  $\beta_1$ . For each of the examples Ex-5—Ex-7, Algorithm 2 was run for Maxiter = 500, for 100 different initializations around the LAD solution, the best solution was used as an initialization for Algorithm 1. Table 3 presents the results obtained with Algorithm 3 (large-scale).

In addition to the above, we considered a large environmental dataset from the R-package `robustbase` with hourly measurements of NOx pollution content in the ambient air. The dataset has  $n = 8088$  samples with  $p = 4$  covariates (including the intercept). The covariates are square-

root of the windspeed ( $x_1$ ), day number ( $x_2$ ), log of hourly sum of NOx emission of cars ( $x_3$ ) and intercept, with response being log of hourly mean of NOx concentration in ambient air ( $y$ ). We considered three different values of  $q \in \{7279, 6470, 4852\}$  corresponding to the 90th, 80th and 60th quantile respectively. We added a small amount of contamination by changing  $\lfloor 0.01n \rfloor$  sample points according to Item-B in Section 5.1. On the modified dataset we ran three different algorithms: Algorithm 3 (large-scale)<sup>7</sup>, MIO (warm-start) i.e. MIO formulation (2.11) warm-started with Algorithm 3 (large-scale) and the LQS algorithm from the R package MASS. In all the following cases, the MIO algorithm was run for a maximum of two hours. We summarize our findings below:

1. For  $q = 7279$ , the best solution was obtained by MIO (warm-start) in about 1.6 hours. Algorithm 3 (large-scale) delivered a solution with relative accuracy (see (5.1)) 0.39% in approximately six minutes. The LQS algorithm from R-package MASS, delivered a solution with relative accuracy 2.8%.
2. For  $q = 6470$ , the best solution was found by MIO (warm-start) in 1.8 hours. Algorithm 3 (large-scale) delivered a solution with relative accuracy (see (5.1)) 0.19% in approximately six minutes. The LQS algorithm from R-package MASS, delivered a solution with relative accuracy 2.5%.
3. For  $q = 4852$ , the best solution was found by MIO (warm-start) in about 1.5 hours. Algorithm 3 (large-scale) delivered a solution with relative accuracy (see (5.1)) 0.14% in approximately seven minutes. The LQS algorithm from R-package MASS, delivered a solution with relative accuracy 1.8%.

Example ( $n, p, \pi$ ) q		Algorithm Used			
		LQS (MASS)	Algorithm-3	MIO formulation (2.11) (cold-start) (warm-start)	
Ex-5 (2001,10,0.4) q=1201	Accuracy Time (s)	65.125 (2.77) 0.30	0 ( 0.0 ) 13.75	273.543 (16.16) 200	0.0 (0.0) 100
Ex-6 (5001,10, 0.4) q=3001	Accuracy Time (s)	52.092 ( 1.33) 0.69	0.0 205.76	232.531 (17.62) 902	0.0 (0.0) 450.35
Ex-7 (10001,20, 0.4) q=6001	Accuracy Time (s)	146.581 (3.77) 1.80	0.0 (0.0) 545.88	417.591 (4.18) 1100	0.0 (0.0) 550

TABLE 3

Table showing performances of various Algorithms for the LQS problem for different moderate-scale examples as described in the text. For each example, “Accuracy” is Relative Accuracy (see (5.1)) the numbers within brackets denote the standard errors; the lower row denotes the averaged cpu time (in secs) taken for the Algorithm. All results are averaged over 20 random examples.

**6. Conclusions.** In this paper, we proposed algorithms for LQS problems based on a combination of first order methods from continuous optimization and mixed integer optimization. Our key conclusions are:

1. The MIO algorithm with warm start from the continuous optimization algorithms solves to provable optimality problems of small ( $n = 100$ ) and medium ( $n = 500$ ) size problems in under two hours.

<sup>7</sup>In this example, we initialized Algorithm 2 with the best Chebyshev fit from forty different subsamples. Algorithm 2 was run for Maxiter=500, with five hundred random initializations. The best solution was taken as the starting point of Algorithm 3

2. The MIO algorithm with warm starts finds high quality solutions for large ( $n = 10,000$ ) scale problems in under two hours outperforming all state of the art algorithms that are publicly available for the LQS problem. For problems of this size, the MIO algorithm does not provide a certificate of optimality in a reasonable amount of time.
3. There exists an optimal solution for the LQS problem for *any* dataset, where the data-points  $(y_i, \mathbf{x}_i)$ 's are not necessarily in general position. Our MIO formulation leads to a simple proof of the breakdown point of the LQS objective value that holds for general datasets and our framework can easily incorporate extensions of the LQS formulation with polyhedral constraints (stemming from regularizers) on the regression coefficient vector.
4. Our view of tractability is not polynomial time solution times. Under such a definition, the Simplex method for linear optimization would not be deemed tractable, which is counter to the empirical evidence for the simplex algorithm. Rather we define tractability the ability of a method to solve problems of practical interest and in times that are appropriate for the application addressed. The MIO approach provides provably optimal solutions to problems of realistic size in reasonable times from an empirical perspective. Modern MIO approaches achieve this by intelligent enumeration (branch and bound and branch and cut). We do not provide polynomial complexity guarantees as these guarantees do not exist unless  $P=NP$ .

*Acknowledgements.* We would like to thank the Associate editor and the reviewers for several insightful comments.

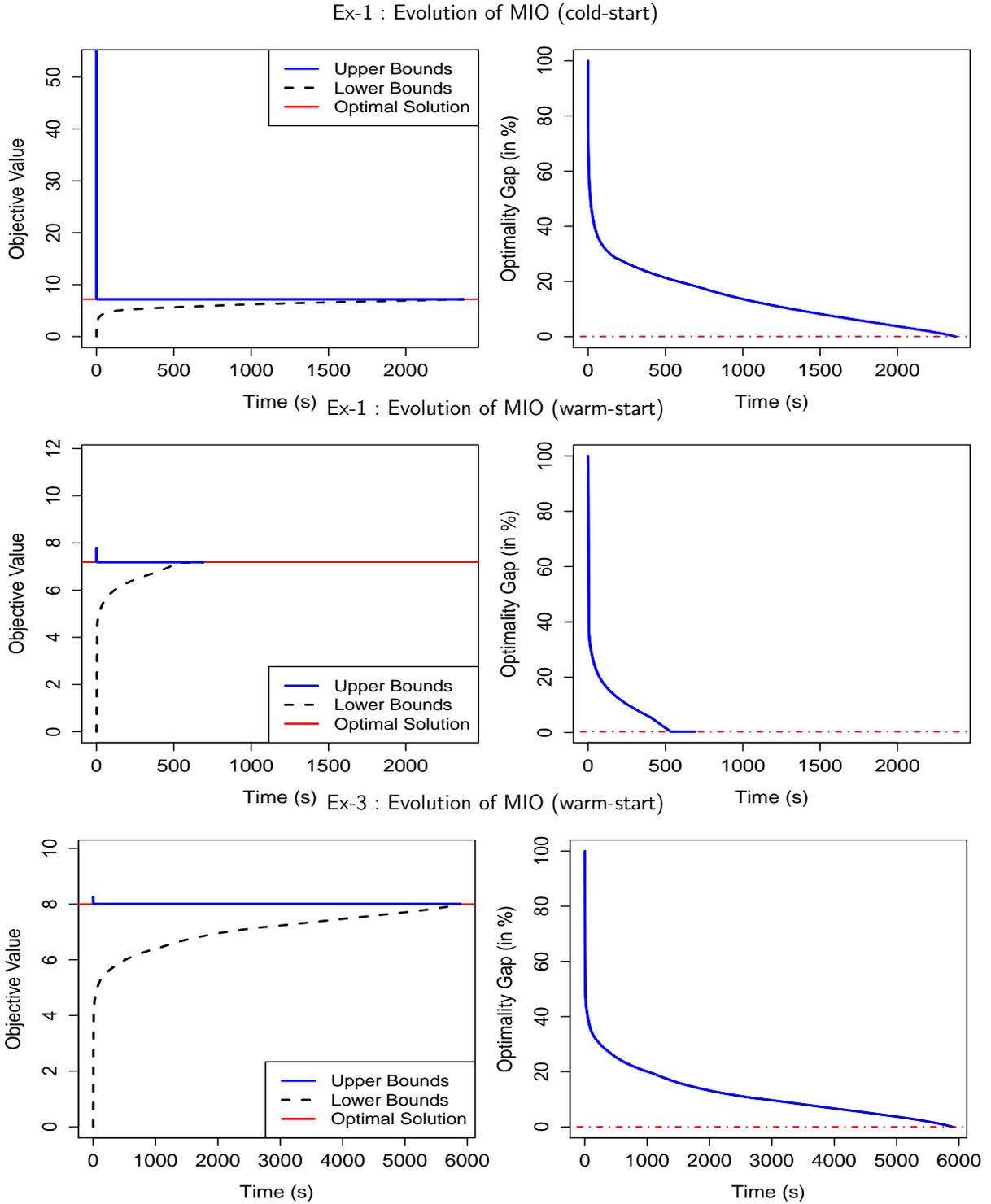


FIG 4. Figure showing evolution of MIO in terms of upper/lower bounds [left panel] and Optimality gaps (in %) [right panel]. Top and middle rows display an instance of Ex-1 with  $(n, p, q) = (201, 5, 121)$  with different initializations, i.e., MIO (2.11) (cold-start) and MIO (2.11) (warm-start) respectively. Bottom row considers an instance of Ex-3 with  $(n, p, q) = (501, 5, 301)$ .

## References.

- AGULLO, J. (1997). Exact Algorithms for Computing the Least Median of Squares Estimate in Multiple Linear Regression. *Lecture Notes-Monograph Series* **31** 133-146.
- BARRETO, H. and MAHARRY, D. (2006). Short Communication: Least median of squares and regression through the origin. *Computational Statistics and Data Analysis* **50** 1391-1397.
- BERNHOLT, T. (2005a). Robust estimators are hard to compute Technical Report No. 52/2005, University of Dortmund.
- BERNHOLT, T. (2005b). Computing the least median of squares estimator in time  $O(nd)$ . In *Proceedings of ICCSA 2005, LNCS 3480* 697-706.
- BERTSIMAS, D. and WEISMANTEL, R. (2005). *Optimization over integers*. Dynamic Ideas Belmont.
- BICKEL, P. J. (1975). One-step Huber estimates in the linear model. *Journal of the American Statistical Association* **70** 428-434.
- BOYD, S. and VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.
- CHAKRABORTY, B. and CHAUDHURI, P. (2008). On an optimization problem in robust statistics. *Journal of Computational and Graphical Statistics* **17** 683-702.
- CLARKE, F. H. (1990). *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, Philadelphia.
- DONOHO, D. L. and HUBER, P. J. (1983). The notion of breakdown point. *A Festschrift for Erich L. Lehmann* 157-184.
- ERICKSON, J., HAR-PELED, S. and MOUNT, D. M. (2006). On the least median square problem. *Discrete and Computational Geometry* **36** 593-607.
- GILONI, A. and PADBERG, M. (2002). Least trimmed squares regression, least median squares regression, and mathematical programming. *Mathematical and Computer Modeling* **35** 1043-1060.
- GUROBI OPTIMIZATION, I. (2013). Gurobi Optimizer Reference Manual.
- HAMPEL, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics* **42** 1887-1896.
- HAMPEL, F. R. (1975). Beyond location parameters: Robust concepts and methods. *Bulletin of the International statistical Institute* **46** 375-382.
- HAWKINS, D. M. (1993). The feasible set algorithm for least median of squares regression. *Computational Statistics and Data Analysis* **16** 81 - 101.
- HAWKINS, D. M., BRADU, D. and KASS, G. V. (1984). Location of several outliers in multiple-regression data using elemental sets. *Technometrics* **26** 197-208.
- HUBER, P. J. (1973). Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics* **1** 799-821.
- HUBER, P. J. (2011). *Robust statistics*. Springer, Berlin.
- HUBERT, M., ROUSSEEUW, P. J. and VAN AELST, S. (2008). High-breakdown robust multivariate methods. *Statistical Science* 92-119.
- MEER, P., MINTZ, D., ROSENFELD, A. and KIM, D. Y. (1991). Robust regression methods for computer vision: A review. *International Journal of Computer Vision* **6** 59-70.
- MOUNT, D. M., NETANYAHU, N. S., PIATKO, C. D., SILVERMAN, R. and WU, A. Y. (2000). Quantile Approximation for Robust Statistical Estimation and k-Enclosing Problems. *International Journal of Computational Geometry and Applications* **10** 593-608.
- MOUNT, D. M., NETANYAHU, N. S., ROMANIK, K., SILVERMAN, R. and WU, A. Y. (2007). A practical approximation algorithm for the LMS line estimator. *Computational Statistics and Data Analysis* **51** 2461 - 2486.
- NESTEROV, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Norwell.
- NUNKESSER, R. and MORELL, O. (2010). An evolutionary algorithm for robust regression. *Computational Statistics and Data Analysis* **54** 3242 - 3248.
- OLSON, C. (1997). An Approximation Algorithm for Least Median of Squares Regression. *Information Processing Letters* **63** 237-241.
- ROCKAFELLAR, R. T. (1996). *Convex Analysis*. Princeton University Press, Princeton.
- ROUSSEEUW, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association* **79** 871-880.
- ROUSSEEUW, P. J. and DRIESSEN, K. (2006). Computing LTS Regression for Large Data Sets. *Data Mining and Knowledge Discovery* **12** 29-45.
- ROUSSEEUW, P. and HUBERT, M. (1997). Recent developments in PROGRESS. In *L1-Statistical Procedures and Related Topics* 201-214.
- ROUSSEEUW, P. J. and LEROY, A. M. (1987). *Robust regression and outlier detection*. Wiley, New York.

- ROUSSEEUW, P. J., DEBRUYNE, M., ENGELEN, S. and HUBERT, M. (2006). Robustness and outlier detection in chemometrics. *Critical Reviews in Analytical Chemistry* **36** 221–242.
- ROUSSEEUW, P., CROUX, C., TODOROV, V., RUCKSTUHL, A., SALIBIAN-BARRERA, M., VERBEKE, T., KOLLER, M. and MAECHLER, M. (2013). *robustbase: Basic Robust Statistics R package version 0.9-10*.
- SHOR, N. Z., KIWIEL, K. C. and RUSZCAYSKI, A. (1985). *Minimization methods for non-differentiable functions*. Springer-Verlag, New York.
- SIEGEL, A. F. (1982). Robust regression using repeated medians. *Biometrika* **69** 242–244.
- SOUVAINE, D. L. and STEELE, J. M. (1987). Time and Space Efficient Algorithms for Least Median of Squares Regression. *Journal of the American Statistical Association* **82** 794–801.
- STEELE, J. and STEIGER, W. (1986). Algorithms and complexity for least median of squares regression. *Discrete Applied Mathematics* **14** 93–100.
- STEWART, C. (1999). Robust Parameter Estimation in Computer Vision. *SIAM Review* **41** 513–537.
- STROMBERG, A. J. (1993). Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression. *SIAM Journal of Scientific Computing* **14** 1289–1299.
- TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* **58** 267–288.
- TIBSHIRANI, R. and TAYLOR, J. (2011). The Solution Path of the Generalized Lasso. *Annals of Statistics* **39(3)** 1335–1371.
- TODOROV, V. and FILZMOSER, P. (2009). An Object-Oriented Framework for Robust Multivariate Analysis. *Journal of Statistical Software* **32** 1–47.

DIMITRIS BERTSIMAS,  
 MIT SLOAN SCHOOL OF MANAGEMENT AND  
 OPERATIONS RESEARCH CENTER,  
 MASSACHUSETTS INSTITUTE OF TECHNOLOGY,  
 CAMBRIDGE, MA.  
 E-MAIL: [dbertsim@mit.edu](mailto:dbertsim@mit.edu)

RAHUL MAZUMDER,  
 DEPARTMENT OF STATISTICS,  
 COLUMBIA UNIVERSITY,  
 NEW YORK, NY.  
 E-MAIL: [rm3184@columbia.edu](mailto:rm3184@columbia.edu)