



Discrete Optimization

SOCP relaxation bounds for the optimal subset selection problem applied to robust linear regression

Salvador Flores^{1,*}

Centro de Modelamiento Matemático (CNRS UMI 2807), Universidad de Chile, Beauchef 851, Santiago, Chile

ARTICLE INFO

Article history:

Received 29 May 2014

Accepted 19 April 2015

Available online 24 April 2015

Keywords:

Global optimization

Integer programming

High breakdown point regression

Branch and bound

Relaxation–linearization technique

ABSTRACT

This paper deals with the problem of finding the globally optimal subset of h elements from a larger set of n elements in d space dimensions so as to minimize a quadratic criterion, with a special emphasis on applications to computing the Least Trimmed Squares Estimator (LTSE) for robust regression. The computation of the LTSE is a challenging subset selection problem involving a nonlinear program with continuous and binary variables, linked in a highly nonlinear fashion. The selection of a globally optimal subset using the branch and bound (BB) algorithm is limited to problems in very low dimension, typically $d \leq 5$, as the complexity of the problem increases exponentially with d . We introduce a bold pruning strategy in the BB algorithm that results in a significant reduction in computing time, at the price of a negligible accuracy lost. The novelty of our algorithm is that the bounds at nodes of the BB tree come from pseudo-convexifications derived using a linearization technique with approximate bounds for the nonlinear terms. The approximate bounds are computed solving an auxiliary semidefinite optimization problem. We show through a computational study that our algorithm performs well in a wide set of the most difficult instances of the LTSE problem.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

In this article we deal with a nonlinear subset selection problem arising in the computation of linear regression estimators with strong robustness properties.

Before entering into the details of the problem we introduce the problem of robust estimation through an example from [Rousseeuw and Leroy \(1987\)](#). In [Fig. 1](#) (left) we show, for each year from 1950 to 1973, the number of outgoing international phone calls from Belgium. The bulk of the data follows a linear model; nonetheless, there are 6 observations that deviate from the majority. In fact, during the period between 1964 and 1969, there was a change on the record system, which actually recorded the total duration, in minutes, of the international phone calls instead of the number of calls. We plot the regression line obtained by a robust method (solid line) and that obtained by the method of least squares (dashed line). The least squares estimation is strongly affected by the outliers.

A more appealing example ([Jalali-Heravi & Konouz, 2002](#)) where a subpopulation acts differently, is shown in [Fig. 1](#) (right). There are plotted, for 32 chemical compounds, a quantity called Krafft point versus a molecular descriptor called heat of formation. There is a

main group that follows a regression line, correctly estimated by a robust estimator (solid line); there is also, besides some few outliers at right, a second, smaller group forming what seems to be another regression line. The observations in the second group correspond to sulfonates. The least squares estimator (dashed line) is not helpful to detect the presence of the second group.

As the reader can figure out, in higher dimensions, where visual inspection is not longer an alternative for detecting outliers, specifically suited methods are needed to deal with outliers. This is what robust estimators are about. Besides robustness, in a sense to be specified soon, robust regression estimators satisfy statistical properties such as asymptotic normality, square-root rate of convergence and equivariance properties ([Maronna, Martin, & Yohai, 2006](#)). Unfortunately, the use of robust estimators is not as widespread as one may expect because their computation is very time-consuming. Unlike other problems arising in Statistics, the difficult problems involved in computing robust estimators remain almost unknown to O.R. specialists.

1.1. Description of the motivating problem

We have at our disposal a sample consisting of n observations of the explicative variables $\{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$. To each observation of explicative variables it corresponds a response or dependent variable, gathered together in a vector $y = (y_1, \dots, y_n) \in \mathbb{R}^n$. We assume that the random variables x and y are related through a linear model, which

* Tel.: +56 22 97 80 611; fax: +56 26 88 97 05.

E-mail address: sflores@dim.uchile.cl

¹ Supported by Fondecyt under grant 3120166.

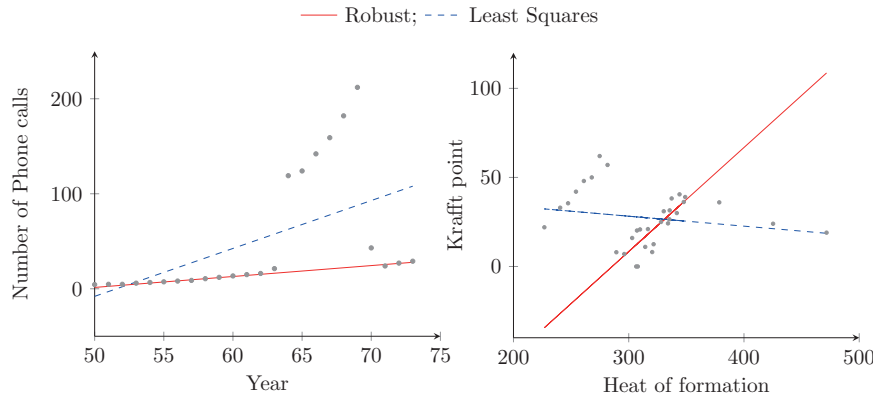


Fig. 1. Some real data containing outliers. At the left: phone calls from Belgium; at the right: Krafft point of chemical compounds.

implies the existence of a vector $\beta \in \mathbb{R}^d$ such that

$$y_i = x_i^\top \beta + \delta_i, \tag{1}$$

with δ_i i.i.d., $E[\delta_i] = 0$, $Var[\delta_i] = \sigma^2$. The objective of linear regression is to estimate the parameter β .

For convenience, we put the explicative variables as rows of a matrix $X \in \mathbb{R}^{n \times d}$,

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(d)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(d)} \\ \vdots & \vdots & \vdots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(d)} \end{pmatrix}.$$

For $\beta \in \mathbb{R}^d$, we denote by $r(\beta)$ the vector of residuals $r = y - X\beta$, with components $r_i = y_i - x_i^\top \beta$. The Least Squares (LS) estimator is obtained by minimizing the sum of the squared residuals:

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^n r_i(\beta)^2.$$

Hereafter we adopt the robustness notion introduced by Donoho and Huber (1983), which is based on the concept of Breakdown Point (BDP). The BDP of an estimator on a sample is defined as the minimum fraction of observations that need to be replaced by arbitrary ones for the estimator to take on arbitrary values. The BDP of the common LS estimator is $1/n$, since it suffices to control just one observation to make the estimator divergent. Therefore, the asymptotic BDP of the LS estimator as n grows to infinity is 0 percent. The same is true if the LS estimator is replaced by any estimator obtained by minimizing a convex function of the residuals. Since the pioneer Least Median of Squares (LMS) estimator (see Rousseeuw and Leroy, 1987, for a precise description), there has been a continuous improvement leading to high BDP estimators with optimal statistical properties, such as asymptotic normality, speed of convergence and efficiency. In this article we focus on the Least Trimmed Squares estimator, which has the best statistical properties and is defined through a well structured optimization problem. Let h be an integer number comprised between $n/2$ and n , and denote by $|r|_{1:n} \leq |r|_{2:n} \leq \dots \leq |r|_{n:n}$ the residuals, ordered by increasing absolute value. The Least Trimmed Squares (LTS) estimator is defined as a solution of the problem:

$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^h r(\beta)_{i:n}^2. \tag{LTS}$$

In words, the LTSE is the vector of regression coefficients $\hat{\beta}$ that minimizes the sum of the h smallest squared distances from the hyperplane defined by $\hat{\beta}$ to the observations y_i . The LTSE attains the

maximum asymptotic BDP of 50 percent by taking $h = \lfloor n/2 \rfloor + \lfloor (d + 1)/2 \rfloor$ (Rousseeuw & Leroy, 1987).

1.2. State of the art

The first approaches to the optimal subset selection problem appeared in the field of pattern recognition (Chen, 2003; Narendra & Fukunaga, 1977; Somol, Pudil, & Kittler, 2004; Yu & Yuan, 1993). Unlike robust regression, the feature selection problem addressed there is a maximization problem, whose difficulties are somehow different from ours. To the best of our knowledge, the LTSE is the only reported application of subset selection involving minimization.

The exact computation of high-BDP estimators for d greater than, say, 5 is a difficult global optimization problem (Bernholt, 2005; Erickson, Har-Peled, & Mount, 2006; Mount, Netanyahu, Piatko, Silverman, & Wu, 2014). Indeed, Erickson et al. (2006) provide results suggesting that any exact algorithm for the related LMS requires, for large n , a time superior to $k \cdot n^d$ for some constant $k > 0$. Mount et al. (2014) extend this result to the LTSE under consideration here proving that, up to a constant, the time required for computing the LTSE for a given coverage level h must be, for large n , bounded between $(n/h)^d$ and n^{d+1} .

For this reason, the overwhelming majority of the literature on computation of robust estimators is focused on stochastic approximation algorithms. Most of these algorithms are constructed upon the basic resampling algorithm proposed originally by Rousseeuw and Leroy (1987) for computing the LMS estimator. Rousseeuw and Van Driessen (2006) developed a refined version including local improvements and adapted to the LTSE. Recently, Torti, Perrotta, Atkinson, and Riani (2012) conducted a benchmark of stochastic algorithms for approximating high-BDP linear regression estimators. The interested reader can find in that article an up-to-date account of stochastic approximation algorithms for robust (though not high-BDP) linear regression. Stochastic approximation algorithms typically provide good approximations to the actual estimator for problems with a number of observations in the hundreds or even in the few thousands. However, they have some shortcomings as well. Two different runs or different implementations of the algorithm may give different results. Also, as it is not guaranteed that the obtained solution is a global minimum, nothing can be said about the actual breakdown point of such approximations. Even a deterministic, constant-factor approximation to a high breakdown point estimator may have a very low breakdown point.

That being said, for small or medium-sized datasets one may be disposed to invest more time to have a guaranteed global solution in return. Unfortunately, despite the notable literature dedicated to stochastic approximations there exist very few deterministic algorithms for computing robust estimators, exactly or approximately. We

can mention the proposals by [Steele and Steiger \(1986\)](#) and [Stromberg \(1993\)](#) for computing the LMS estimator; both based on enumeration of elemental subsets. For the particular case of LMS regression with two predictors ($d = 2$), [Mount, Netanyahu, Romanik, Silverman, and Wu \(2007\)](#) devised a Branch and Bound algorithm with an asymptotic running time of $O(n \log_2 n)$.

The first great step forward in the computation of the LTSE came with the Branch and Bound algorithm (BBA) of [Agulló \(2001\)](#). The BBA is an adaptation for minimization problems of the ‘feature subset selection’ branch and bound maximization algorithm by [Narendra and Fukunaga \(1977\)](#) and relies on the monotonicity of the problem. At a glance, the BBA enumerates all the subsets of h observations out of the n , by starting from the empty set and adding one observation at a time. Since the sum of the squared residuals increases when an additional observation is added to the LS fit, if a subset of observations with cardinality $k < h$ is found to give a sum of squared residuals larger than that of the incumbent set, then by monotonicity all the sets containing that set can be discarded from further examination. The BBA is reported to be efficient in datasets with up to about $n = 50$ observations and $d = 5$ features. To a large extent, the difficulty of the BBA in tackling larger problems stems from the following limitations of the monotonicity bound

- It does not provide useful information for small subsets: since in dimension d it is always possible to fit d observations exactly, it is impossible to obtain a non-trivial (positive) lower bound for the sum of squared residuals for a fit of h observations containing a given subset of d or less observations. In the enumeration tree, this amounts to not having a lower bound to prune at the top d levels.
- It does not look ahead: for instance, if $n = 15$ and $h = 6$ it gives “the sum of the squared residuals of a regression over six observations comprising observations 2, 5 and 8 is greater than the sum of the squared residuals of the regression over observations 2, 5 and 8” without quantifying the increase in sum of squared residuals due to the incorporation of three more observations.

Consequently, as the dimension of the explicative variables increases, the BBA must examine a large number of elemental subsets, since pruning is possible only at the bottom levels of the enumeration tree, even if a good global upper bound is available beforehand. On the other hand, the BBA uses a quite efficient explicit formula for computing the increase in sum of squared residuals when adding one observation, which makes his algorithm quite efficient at the lower levels of the enumeration tree.

[Hofmann, Gatu, and Kontoghiorghe \(2010\)](#) extended the BBA for obtaining the LTSE for many coverage values h at once, besides improving the numerical linear algebra used to compute lower bounds. Very recently, [Bertsimas and Mazumder \(2014\)](#) proposed a linear Mixed Integer Optimization (MIO) formulation of the Least Quantile of Squares (and in particular the LMS) regression problem. They report good results at solving to provable optimality problems of small ($n = 100$) and medium ($n = 500$) size for $d = 3$. Some impressive results are reported for approximate (albeit deterministic) solutions for problems with $d \leq 20$ and n in the order of 10^4 . The BDP of the regression performed with approximate solutions is not reported.

1.3. Innovations and contributions

We model the computation of the LTSE as a nonconvex optimization problem comprising continuous and binary variables with nonlinear interdependence. Since, the nonlinear coupling of the variables makes obtaining bounds on the continuous variables impractical, we devise a technique to obtain approximate bounds on the continuous variables; this is done by solving a Semi-Definite Programming (SDP) problem only once. Then we use the approximate bounds to obtain, via the Relaxation-Linearization Technique (RLT), a Second Order

Cone Programming (SOCP) problem whose solution approximates the solution of the original nonconvex problem. Finally, the SOCP approximations are carefully used to obtain useful lower bounds at the top levels of the subset enumeration tree, where existing algorithms just pass through.

1.4. Outline of the paper

We begin by showing, in [Section 2](#), the alternative formulations of the problem that permit to get rid of the order statistics involved in Problem (LTS). More specifically, we show that the problem can be cast as a nonlinear mixed-integer program or a concave program, both particular cases of the best subset selection problem. Then, in [Section 3](#) we introduce approximate convexifications for products involving binary variables and nonlinear continuous terms when an upper bound for the continuous variables is not available. We analyse the validity of the relaxation obtained using an estimation of the bound on the continuous variables and their applicability in a branch and bound algorithm. In [Section 4](#) we show how to obtain an estimation of the bound on the continuous variables to be used for the approximate convexification using a known SDP reformulation of the concave maximization problem. [Section 5](#) describes the actual implementation of a branch and bound algorithm incorporating bounds from approximate convexifications. In [Section 6](#) we present the results of a computational study showing the performance gains in a branch and bound algorithm due to SOCP bounds in a large set of problems. Finally, [Section 7](#) concludes the paper with a discussion on further avenues for research in this subject.

2. Formulation as a best subset problem

Problem (LTS) can be written as a mixed integer nonlinear program using the fact that for arbitrary $r \in \mathbb{R}^n$, if $r_{1:n} \leq r_{2:n} \leq \dots \leq r_{n:n}$ denote its ordered components, then

$$\sum_{i=1}^h r_{i:n} = \min \left\{ \sum_{i=1}^n w_i r_i \mid w \in \{0, 1\}^n, \sum_{i=1}^n w_i = h \right\},$$

and

$$\sum_{i=1}^h r(\beta)_{i:n}^2 = \min_{w \in C_h} \sum_{i=1}^n w_i r_i(\beta)^2,$$

where $C_h = \{w \in \{0, 1\}^n, \sum_{i=1}^n w_i = h\}$ is a representation of all the subsets of $\{1, \dots, n\}$ of size h .

Therefore our original Problem (LTS) is equivalent to the following mixed-integer nonlinear programming problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^n w_k r_k^2, \\ \text{s.t.} \quad & r + X\beta = y, \\ & e^\top w = h, \\ & w \in \{0, 1\}^n, \beta \in \mathbb{R}^d, r \in \mathbb{R}^n, \end{aligned} \tag{2}$$

where e is the $n \times 1$ vector of ones.

By splitting the variables of Problem (2) we see that it is equivalent to

$$\min \{ v(w) \mid w \in C_h \}, \tag{3}$$

where $v(w)$ is the value of the weighted LS problem

$$v(w) = \inf \left\{ \sum_{k=1}^n w_k r_k^2 : \beta \in \mathbb{R}^d, r = y - X\beta \right\} \tag{4}$$

obtained by minimizing over β and r for fixed $w \in C_h$ in (2). Hence, Problem (2) amounts to selecting the subset of h observations with the least sum of squared residuals. The function v defined in (4)

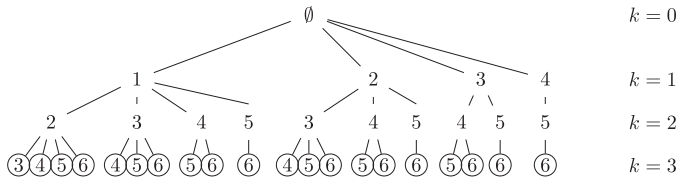


Fig. 2. The BB tree for $n = 6$ and $h = 3$.

is concave, therefore Problem (LTS) can be thought as a concave minimization problem. Giloni and Padberg (2002) were the first to show this property, and used it to devise a local minimization procedure. Nguyen and Welsch (2010) revisited this formulation and derived an SDP formulation of the corresponding maximization problem. Unfortunately, the degeneracy of the feasible domain makes it difficult to apply concave minimization algorithms to problem (LTS). In this paper we tackle the problem in the form given in (2).

2.1. The enumeration tree

In Fig. 2 we depict the enumeration tree constructed by the branch and bound algorithm, in a small example with $h = 3$ and $n = 6$.

The circled nodes are the leaves; each leaf represents a subset of 3 observations (an element of C_h), which is obtained adding recursively the parent of each node until the root \emptyset is reached. For example, at the end of the second branch from right to left there are two leaves, the leaf at the right is associated to the subset of observations 6, 4 and 3, and that at left to observations 5, 4 and 3. In terms of the optimization variable w , they are associated to the points $(0, 0, 1, 1, 0, 1)^T$ and $(0, 0, 1, 1, 1, 0)^T$ respectively. Any node has associated two index sets S_0 and S_1 representing the variables fixed to 0 and 1 respectively, each of cardinality J_0 and J_1 . Using these two quantities we can compute the number of child nodes as $n - J_0 - h + 1$ and the number of leaves that can be reached from the l th child node as $\binom{n - J_1 - J_0 - l}{h - J_1 + 1}$.

2.2. The monotonicity lower bound

The value of the function v at a point $w \in [0, 1]^n$ gives the least sum of weighted squares of residuals with weights w . In particular if $w \in \{0, 1\}^n$, $w_i = 1$ for $i \in J \subseteq \{1, \dots, n\}$ and $w_i = 0$ for $i \notin J$, then $v(w)$ is the sum of squares of the fit to the subset of observations J . The value of $v(w)$ is finite as long as the matrix $M(w) = X^T D(w) X$ is invertible, and in this case (Agulló, 2001),

$$v(w + e_j) = v(w) + \frac{r_j(w)^2}{1 + x_j^T M(w)^{-1} x_j}, \tag{5}$$

where $D(w)$ is the diagonal matrix formed from the vector w , $r_j(w)$ is the j th residual obtained from the weighted least squares problem (4) with weights w , and e_j is the j th euclidean basis vector. Formula (5) gives the change in the sum of squared residuals by adding observation j to the fit J , provided that J contains at least d linearly independent x_i s. Note that $v(w + e_j) - v(w) \geq 0$, which means that the objective function is non-decreasing from one node to any of its children. Therefore, the RSS at one node is a lower bound for the objective function of its children. If the tree is examined using a depth-first search strategy, it is possible to keep a Cholesky factorization of $M(w)$ from which to perform rank-one updates in order to quickly compute the quantities $x_j^T M(w)^{-1} x_j$. The shorthands of the monotonicity bound were already mentioned: Formula (5) requires $M(w) = X^T D(w) X$ to be invertible, thus it is not applicable at the d top levels of the tree, and it provides loose bounds when J_1 is far from h (when there are few more than d observations).

3. Pseudo-convexifications with approximate bounds

At each node of the tree we need to (under) estimate the value of the problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^n w_k r_k^2, \\ \text{s.t.} \quad & r + X\beta = y, \\ & e^T w = h, \\ & 0 \leq w \leq 1, \\ & \beta \in \mathbb{R}^d, \quad r \in \mathbb{R}^n \end{aligned} \tag{6}$$

with the additional constraints

$$\begin{aligned} w_k &= 1, \quad k \in S_1 \\ w_k &= 0, \quad k \in S_0 \end{aligned} \tag{7}$$

for two index sets S_1, S_0 particular to each node. Suppose that we had an upper bound Π_k for each quadratic term r_k^2 ; then we can apply a linearization technique (Adams, Forrester, & Glover, 2004; Adams & Sherali, 1990; 1993; Glover, 1975) to get rid of the product of the continuous term r_k^2 with the binary variable w_k . This is done by constructing a new continuous variable u_k to replace each product $w_k r_k^2$, and adding a number of additional constraints in order to ensure that the value of the variable u_k equals the product $w_k r_k^2$ at any feasible point of the new problem. Applied to Problem (6) this procedure yields to the following SOCP problem

$$\begin{aligned} \min \quad & \sum_{k=1}^n u_k \\ \text{s.t.} \quad & r_k^2 - \Pi_k(1 - w_k) \leq u_k, \quad 1 \leq k \leq n \\ & w_k = 1, \quad k \in S_1 \\ & w_k = 0, \quad k \in S_0 \\ & \langle e, w \rangle = h \\ & r + X\beta = y, \\ & u \in \mathbb{R}_+^n, \quad r \in \mathbb{R}^n, \quad \beta \in \mathbb{R}^d, \quad w \in \{0, 1\}^n. \end{aligned} \tag{P}$$

From the standard theory of Glover (1975), if $\Pi_k \geq \bar{r}_k^2$, where \bar{r} are the residuals at a solution to Problem (LTS), then Problem (P) coincides with (LTS) for any binary realization of w . Unfortunately, as we show in the following section, the nonlinear coupling of the variables restrains us from efficiently obtaining a guaranteed upper bound for the residuals of Problem (LTS). For this reason we introduce pseudo-convexifications, which are instances of problem (P) for an approximate upper bound $(\Pi_k)_{1 \leq k \leq n}$. We say that a solution to (P) is consistent if $r_k^2 < \Pi_k$ for any $k = 1, \dots, n$ at the optimal r . Consistency of the solution to a pseudo-convexification is a necessary condition for being an actual convexification, but in general it is not sufficient. As a consequence, using bounds obtained from a pseudo-convexification in a branch and bound algorithm can lead to pruning branches potentially yielding a new solution. Of course, this drawback can be avoided by giving very large values to Π . Nevertheless, larger values of Π yields to weaker lower bounds when relaxing the binary constraint. Therefore, we are led to find a compromise between a large bound ensuring the equivalence of problems (LTS) and (P), and a smaller bound promoting tight lower bounds for Problem (6). In the next section we describe a strategy for obtaining bounds achieving that compromise.

4. Obtaining good approximate bounds

It is customary when linearizing polynomial programs to suppose that the optimization takes place on a bounded separable polytope, and that upper and lower bounds for each variable exists and can be computed by linear programming. This does not hold in our case. In our problem the variable w ranges over the unit cube in \mathbb{R}^n , and all the other variables have a nonlinear dependency on w . Indeed, for each

w there exists a unique feasible β_w , as seen from (4); even more, β_w is the unique solution to the system $X^T D(w) X \beta = X^T D(w) y$. The residuals are linked to β , and therefore to w , by the linear constraint $r + X\beta = y$. Therefore, an exact upper bound for r_k^2 can be obtained by solving the following auxiliary problem

$$\begin{aligned} \max_w & r_k^2 \\ \text{s.t.} & e^T w = h \\ & r + X\beta = y, \\ & X^T D(w)r = 0 \\ & 0 \leq w \leq 1. \end{aligned} \tag{A_k}$$

Problem (A_k) is a maximization problem on w only but, unlike Problem (3), it is not a concave maximization problem. Even it it could be done efficiently, solving n problems would be cumbersome. For that reason we look for one single bound for all the residuals that can be efficiently computed. A closely related problem is that of maximizing the weighted sum of squared residuals, which amounts to maximizing the function v defined in (4),

$$\begin{aligned} \max & v(w) \\ \text{s.t.} & e^T w = q, \\ & 0 \leq w \leq 1, \end{aligned} \tag{8}$$

for some $1 \leq q \leq n$. This is a concave maximization problem, therefore a global minimum can be efficiently computed. In fact, [Nguyen and Welsch \(2010\)](#) showed that Problem (8) can be cast as an SDP problem, therefore it can be solved using standard widely-available software. If $d + 1 \leq q \leq n$ and we force the variables to be binary, the solution to Problem (8) is the subset of q observation with the largest sum of squared residuals. In any case, the value of the problem is monotone non-decreasing for $1 \leq q \leq n$. Moreover, it has the property of going to $+\infty$ if any subset of the observations is replaced by divergent ones. After extensive numerical experiments we found that solving (8) for $q = d/2$ yields to an effective approximation of the upper bound for the residuals.

5. A branch and bound algorithm with SOCP bounds (S-BB)

In our implementation, the Narendra–Fukunaga tree described in [Section 2.1](#) is examined using a depth-first search strategy. We perform an in-level node ordering to take advantage of the unbalanced structure of the tree, as leftmost branches have many more children than those at the right. The innovations of our algorithm take place at the d top levels of the tree; at level $d + 1$ and below the algorithm behaves like the BBA of [Agulló \(2001\)](#).

5.1. Preliminaries

We used the LS estimator as an initial solution; as a consequence, our algorithm is deterministic. As indicated in [Section 4](#), we set Π equal to the optimal value of Problem (8) with $q = d/2$. Problem (8) is solved using the SDPT3 interior-point solver ([Tütüncü, Toh, & Todd, 2003](#)).

5.2. Lower bounds

Problem (P) with the binary constraint relaxed to $0 \leq w_k \leq 1$ is denoted as (\bar{P}) . The quadratic constraint $r_k^2 - \Pi(1 - w_k) \leq u_k$ can be cast as a SOCP constraint, for that reason we call Problem (\bar{P}) the SOCP relaxation. Problem (\bar{P}) was solved using CPLEX 12.5. At each node we first check the size of the subtree below each child node. Even nodes at top levels of the tree can have very few leaves below, in which case it is not worth spending time solving the SOCP relaxation. We launch the SOCP relaxation only for children with more than 10^6 leaves. If the solution to Problem (\bar{P}) results to be consistent, and the value of

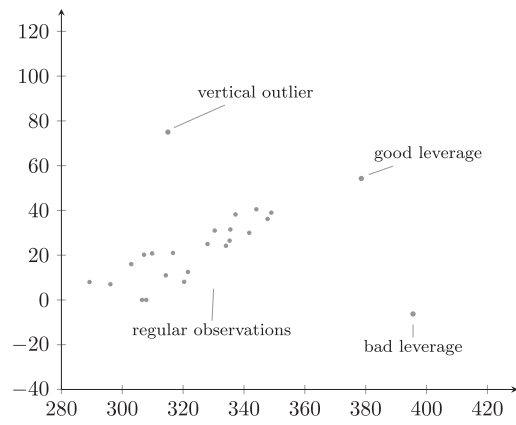


Fig. 3. A taxonomy of regression outliers.

the problem is greater than the current upper bound, the branch is pruned. Otherwise, the residuals of the solution are still useful for ranking the children and performing the in-level ordering, by putting the observations with the largest residuals at the left, to promote subsequent pruning of large branches.

5.2.1. Adjusting the bounds on r

A look at problem (P) shows that the optimal values of Π_k can be anticipated for $k \in S_0 \cup S_1$. For $k \in S_1$ the upper bound does not enter into play, we always have $u_k = w_k r_k^2$; on the contrary, for $k \in S_0$ we should always have $u_k = 0$, therefore we set $\Pi_k = 10 \cdot \Pi$, where Π is the upper bound obtained by solving Problem (8) and used for $k \notin S_0 \cup S_1$. In practice this forces $u_k = 0$, and does not spoil the conditioning of the SOCP problems as a huge number would do (in theory, we should set those Π_k to $+\infty$).

5.3. Local improvements

Another innovation of our BB algorithm is the incorporation of a local search. Each time a leaf is examined, we apply the *concentration steps* of [Rousseeuw and Van Driessen \(2006\)](#) to obtain an eventually better incumbent solution, and use it to update the global upper bound of the algorithm.

6. Computational study

Now we illustrate through a computational study the impact of incorporating the SOCP lower bound in a branch-and-bound algorithm. In order to perform a systematic study, we generated synthetic datasets with sizes in a controlled range. It is known that the structure of the outliers, and not only their magnitudes, strongly affects the regression technique as well as the behaviour of the approximating algorithm. In [Fig. 3](#) we illustrate the taxonomy of linear regression outliers ([Rousseeuw & van Zomeren, 1990](#)). Outliers are called vertical if only the y component (the response) is contaminated. Vertical outliers are the more benign ones, and even some convex estimators, such as the ℓ_1 estimator, can cope with them to some extent ([Giloni & Padberg, 2004](#)). Leverage points are points whose explicative variables are corrupted. In contrast to vertical outliers, leverage outliers can be very harmful. Excepting the case of the “good leverage points” illustrated in [Fig. 3](#), which are in general not considered as outliers, bad leverage points are the most adversarial type of contamination.

For our study we generate synthetic data with outliers in the following way:

- We generate regular observations following model (1), with δ standard normal, for different number of cases (n) and explicative variables (d).

Table 1
Computing times (in tens of seconds) for high leverage outliers.

Alg.	n	d						
		12	13	14	15	16	17	18
BBA	30	0.51	0.35	0.51	0.26	0.36	0.14	0.18
	35	5.19	4.79	7.99	5.68	8.72	5.02	7.29
	40	22.15	23.68	40.85	33.59	56.63	38.99	59.27
S-BB	30	0.49	0.34	0.51†	0.26	0.35	0.14	0.18
	35	4.60	4.21	6.80	5.05†	7.65†	4.45	6.41†
	40	22.53	20.89	34.90	26.91	45.44	29.87	46.49

Table 2
Computing times (in tens of seconds) for heavy tail outliers.

Alg.	n	d						
		12	13	14	15	16	17	18
BBA	30	0.52	0.35	0.52	0.26	0.35	0.14	0.19
	35	5.26	4.81	8.07	5.67	8.69	5.03	7.29
	40	21.41	23.90	41.41	34.82	56.57	39.71	59.47
S-BB	30	0.50	0.34	0.52	0.26†	0.35	0.14	0.19
	35	4.52	4.21	6.86	4.83	7.71	4.48†	6.53
	40	22.11	21.43	34.70	27.46	44.54†	31.44	46.72

Table 3
Success rates of the S-BB algorithm.

Percent	n			Average
	30	35	40	
High leverage outliers	99.55	98.66	100	99.40
Heavy tail outliers	99.55	99.55	99.55	99.55
Average	99.55	99.10	99.77	99.47

- On each dataset we replace 10 regular observations by bad leverage outliers.

The bad leverage outliers were obtained by shifting randomly selected observations in two different ways: by a large, deterministic shift (high leverage outliers) and by adding a random term drawn from a Laplace distribution (heavy tail outliers).

For each combination n/d /type-of-contamination we drew 25 datasets as described above and measured the total time spent by the S-BB algorithm described in Section 5 and by the BBA (Agulló, 2001) in computing the LTSE with a breakdown point of 50 percent ($h = \lfloor n/2 \rfloor + \lfloor (d+1)/2 \rfloor$).

All computations were done in MATLAB version R2008b on a 64-bit Linux machine, with 8 cores and 6 Gigabyte RAM. For the SOCP relaxations we used IBM ILOG CPLEX Studio v. 12.5 via its MATLAB interface.

The computing times in tens of seconds, averaged over the 25 repetitions, are shown in Tables 1 and 2 respectively. The impact of the SOCP bounds is, as expected, more important as the number of explicative variables increases, and more pronounced for larger n . The reduction in computing time exceeds the 20 percent for $n = 40$ and d greater than 15. The accuracy of the solution is largely preserved; in Table 3 we show the rate of success, which is larger than 99 percent in all but one of the cases. The computing times in Tables 1 and 2 marked with a dagger are averages excluding the run that did not give the exact solution.

Further reductions in computing time are possible by relaxing the optimality goal. In this direction, we can mention that the fraction of the SOCP relaxations resulting in inconsistent solutions is not negligible; using those solutions to derive bounds could result in a great performance improvement. Another way to do the same thing is by decreasing the parameter q used to obtain the approximate bound Π . However, the goal of this work was to improve the computation of the LTSE with the least possible lost in accuracy, and it was achieved.

7. Conclusions and perspectives

We have presented an approximate convex relaxation for the LTS problem. Its incorporation in a branch-and-bound algorithm yields to significant savings in computing time at the price of a negligible accuracy lost. Our standpoint is that of improving the computation of an estimator with well-studied statistical properties. Now then, with the understanding of the underlying problem gained with this study one could propose alternative techniques yielding to robust estimators defined with an eye on computability. Concretely, think of the estimator defined as the solution to Problem (P), with Π obtained by solving (8) for some $1 \leq q \leq n$. If the binary constraint is relaxed, such an estimator is defined by two convex optimization problems, with a range of applicability in the thousands of observations; if the binary constraint is kept, we dispose of SOCP convex relaxations at any node of the BB tree, without wasting time with inconsistent relaxations. The study of the statistical and computational aspects of that proposal will be the subject of a future work.

References

- Adams, W. P., Forrester, R. J., & Glover, F. W. (2004). Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. *Discrete Optimization*, 1(2), 99–120.
- Adams, W. P., & Sherali, H. D. (1990). Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2), 217–226.
- Adams, W. P., & Sherali, H. D. (1993). Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(3), 279–305.
- Agulló, J. (2001). New algorithms for computing the least trimmed squares regression estimator. *Computational Statistics & Data Analysis*, 36(4), 425–439.
- Bernholt, T. (2005). Computing the least median of squares estimator in time $O(n^d)$. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Lagana, H. P. Lee, Y. Mun, D. Taniar, & C. Tan (Eds.), *Computational science and its applications – ICCSA 2005*. In *Lecture Notes in Computer Science: vol. 3480* (pp. 697–706). Berlin Heidelberg: Springer.
- Bertsimas, D., & Mazumder, R. (2014). Least quantile regression via modern optimization. *The Annals of Statistics*, 42(6), 2494–2525.
- Chen, X. (2003). An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24(12), 1925–1933.
- Donoho, D., & Huber, P. J. (1983). The notion of breakdown point. In *A Festschrift for Erich L. Lehmann*. In *Wadsworth Statist./Probab. Ser.* (pp. 157–184). Wadsworth.
- Erickson, J., Har-Peled, S., & Mount, D. M. (2006). On the least median square problem. *Discrete Computational Geometry*, 36(4), 593–607.
- Giloni, A., & Padberg, M. (2002). Least trimmed squares regression, least median squares regression, and mathematical programming. *Mathematical and Computer Modelling*, 35(9–10), 1043–1060.
- Giloni, A., & Padberg, M. (2004). The finite sample breakdown point of ℓ_1 -regression. *SIAM Journal on Optimization*, 14(4), 1028–1042.
- Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4), 455–460.
- Hofmann, M., Gatu, C., & Kontoghiorghes, E. J. (2010). An exact least trimmed squares algorithm for a range of coverage values. *Journal of Computational and Graphical Statistics*, 19(1), 191–204.
- Jalali-Heravi, M., & Konouz, E. (2002). Use of quantitative structure-property relationships in predicting the Krafft point of anionic surfactants. *Internet Electronic Journal of Molecular Design*, 1, 410–417.
- Maronna, R. A., Martin, R. D., & Yohai, V. J. (2006). *Robust statistics*. Wiley series in probability and statistics. John Wiley & Sons.
- Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2014). On the least trimmed squares estimator. *Algorithmica*, 69(1), 148–183.
- Mount, D. M., Netanyahu, N. S., Romanik, K., Silverman, R., & Wu, A. Y. (2007). A practical approximation algorithm for the LMS line estimator. *Computational Statistics & Data Analysis*, 51(5), 2461–2486.
- Narendra, P., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26(9), 917–922.
- Nguyen, T. D., & Welsch, R. (2010). Outlier detection and least trimmed squares approximation using semi-definite programming. *Computational Statistics & Data Analysis*, 54(12), 3212–3226.
- Rousseeuw, P., & Van Driessen, K. (2006). Computing LTS regression for large data sets. *Data Mining and Knowledge Discovery*, 12), 29–45.
- Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. John Wiley & Sons.
- Rousseeuw, P. J., & van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*, 85(411), 633–639.
- Somol, P., Pudil, P., & Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 7, 900–912.
- Steele, J. M., & Steiger, W. L. (1986). Algorithms and complexity for least median of squares regression. *Discrete Applied Mathematics*, 14(1), 93–100.

- Stromberg, A. J. (1993). Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression. *SIAM Journal on Scientific Computing*, 14(6), 1289–1299.
- Torti, F., Perrotta, D., Atkinson, A. C., & Riani, M. (2012). Benchmark testing of algorithms for very robust regression: FS, LMS and LTS. *Computational Statistics & Data Analysis*, 56(8), 2501–2512.
- Tütüncü, R. H., Toh, K. C., & Todd, M. J. (2003). Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95, 189–217.
- Yu, B., & Yuan, B. (1993). A more efficient branch-and-bound algorithm for feature-selection. *Pattern Recognition*, 26(6), 883–889.