OR Spectrum
https://doi.org/10.1007/s00291-021-00627-y

ORIGINAL ARTICLE

Leveraged least trimmed absolute deviations

Nathan Sudermann-Merx[1] · **Steffen Rebennack**[2]

Received: 25 March 2020 / Accepted: 18 March 2021

Abstract
The design of regression models that are not affected by outliers is an important task which has been subject of numerous papers within the statistics community for the last decades. Prominent examples of robust regression models are least trimmed squares (LTS), where the k largest squared deviations are ignored, and least trimmed absolute deviations (LTA) which ignores the k largest absolute deviations. The numerical complexity of both models is driven by the number of binary variables and by the value k of ignored deviations. We introduce leveraged least trimmed absolute deviations (LLTA) which exploits that LTA is already immune against y-outliers. Therefore, LLTA has only to be guarded against outlying values in x, so-called leverage points, which can be computed beforehand, in contrast to y-outliers. Thus, while the mixed-integer formulations of LTS and LTA have as many binary variables as data points, LLTA only needs one binary variable per leverage point, resulting in a significant reduction of binary variables. Based on 11 data sets from the literature, we demonstrate that (1) LLTA's prediction quality improves much faster than LTS and as fast as LTA for increasing values of k and (2) that LLTA solves the benchmark problems about 80 times faster than LTS and about five times faster than LTA, in median.

Keywords Trimmed absolute deviations · Least absolute deviations · Least trimmed squares · Robust statistics · Combinatorial machine learning

Mathematics Subject Classification 62J05 · 62-04 · 90C11 · 90C30

✉ Steffen Rebennack
steffen.rebennack@kit.edu

Nathan Sudermann-Merx
nathan.sudermann-merx@basf.com

[1] Department of Data Science for Materials, BASF SE, Ludwigshafen, Germany

[2] Institute for Operations Research (IOR), Stochastic Optimization (SOP), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Published online: 15 April 2021

1 Introduction

1.1 Machine learning, regression and optimization

Machine learning is a fast growing field of research that inherits and combines methods from statistics, computer science and optimization to tackle a vast variety of applications like fraud detection, recommender systems, predictive maintenance and autonomous driving (Marsland 2015). One subfield of machine learning is supervised learning, whose task is to train a function on labeled data. This stands in contrast to other applications, like anomaly detection or clustering, where no labels are available and which are therefore examples of unsupervised learning (Bishop 2006). The most popular examples of supervised learning are classification and regression tasks. While classification aims at assigning discrete values to data points (e.g., binary values for cancer detection), regression methods train functions that assign continuous numbers to data points (e.g., prediction of house prices). Commonly used candidate mappings are (piecewise) linear functions, splines, tree-based models and neural networks (Clark and Pregibon 2015; Goldberg et al. 2021; Krasko and Rebennack 2017; Micula and Micula 2012; Rebennack and Kallrath 2015; Rebennack and Krasko 2020; Specht 1991).

The training procedure involves the minimization of a so-called loss function that measures the distance of the observations to the corresponding predictions. Minimizing the loss function results typically in an unconstrained smooth optimization problem that is tackled by variants of the stochastic gradient descent method which is a lightweight modification of gradient descent where only parts of the gradient are evaluated in each iteration (Schmidt et al. 2017; Robbins and Monro 1951). Other optimization-related topics within machine learning are Bayesian optimization (Snoek et al. 2012) or the optimization of pretrained machine learning models in the feature space (Thebelt et al. 2020b).

Mixed-integer linear optimization (MILO) models involve linear terms in the decision variables as well as integrality restrictions (for some) of the decision variables (Jünger et al. 2009; Wolsey and Nemhauser 1999). A very rich class of optimization problems in practice can be modeled using MILO models. Current state-of-the-art solvers for general MILO models use so-called branch-and-cut algorithms. The idea of branch-and-cut algorithms is to repeatedly solve linear optimization problems (these are easy to solve) which are obtained by relaxing the integrality restrictions on the decision variables. The linear optimization problems are updated by additional restrictions on the relaxed variables in order to cut out fractional values. This is called branching. In the worst case, there are exponentially many such branches in the number of integer variables. The branching is accompanied by cutting planes whose goal is to cut away fractional solutions (without the need to execute the costly branching). Therefore, as a general rule-of-thumb, fewer integer variables lead to lower computational times (though this is not always true). We make use of this observation in this paper.

Dimitris Bertsimas was one of the first researchers to point out that recent advances in linear and quadratic mixed-integer optimization have been rarely

noticed in the statistics and machine learning communities. This inspired him to publish a series of papers under the motto "Machine Learning under a Modern Optimization Lens" that are summarized in the eponymous book (Bertsimas and Dunn 2019). Bertsimas' assessment was confirmed by some of the most renowned researchers in the statistics and machine learning community, Trevor Hastie and Robert Tibshirani, who state (Hastie et al. 2017):

> In exciting new work, Bertsimas et al. (2016) showed that the classical best subset selection problem in regression modeling can be formulated as a mixed integer optimization (MIO) problem. Using recent advances in MIO algorithms, they demonstrated that best subset selection can now be solved at much larger problem sizes that what was thought possible in the statistics community.

This paper was heavily inspired by Bertsimas' observation that mixed-integer optimization is still relatively unknown but can be applied to many optimization problems in the context of machine learning and statistics. Therefore, we present *Leveraged Least Trimmed Absolute Deviations (LLTA)*, a mixed-integer based robust regression model, whose main idea we explain now.

1.2 Motivation

Let $f_\theta : \mathbb{R}^n \to \mathbb{R}$,

$$f_\theta(x) = \theta_0 + \sum_{i=1}^{n} \theta_i x_i$$

be a *linear candidate function* whose parameter $\theta \in \mathbb{R}^{n+1}$ we want to determine optimally with respect to some labeled training data

$$(x^1, y_1), \ldots, (x^j, y_j), \ldots, (x^N, y_N) \in \mathbb{R}^n \times \mathbb{R},$$

with $x^j \in \mathbb{R}^n$ for $j = 1, \ldots, N$. The most popular idea of obtaining such a function f_θ is referred to as *Ordinary Least Squares (OLS)* and goes back to Legendre or Gauß (Stigler 1981) at the end of the 18th century. Let

$$r_{j,\theta} = \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j - y_j$$

be the *residual* of f_θ with respect to the jth data point, $j = 1, \ldots, N$. Then, OLS computes θ by solving the *unconstrained convex quadratic optimization problem*

$$\min_{\theta} \sum_{j=1}^{N} r_{j,\theta}^2. \tag{1}$$

OLS is computationally attractive as it possesses a closed-form solution. However, it is very sensitive with respect to outliers. To soften this sensitivity to

outliers, an alternative approach is to minimize the ℓ_1-norm of the residual vector $r^\theta = (r_{1,\theta}, \ldots, r_{j,\theta}, \ldots, r_{N,\theta})$ instead of the ℓ_2-norm. This results in the *Least Absolute Deviations (LAD)*, a problem that was stated in 1757 by Boscovich (Koenker and Bassett 1985), even before OLS. LAD results in the *unconstrained convex piecewise linear optimization problem*

$$\min_{\theta} \sum_{j=1}^{N} |r_{j,\theta}|. \tag{2}$$

In contrast to (1), LAD does not have a closed-form solution but can be reformulated and solved as a *linear (continuous) optimization problem (LP)* or tackled by a subgradient-based method. When only estimating θ_0 and all $\theta_i = 0$, then this leads to the so-called *location model* (Bassett 1991). For this case, an optimal estimator for θ_0 is simply the median of the sorted data points $y_{(j)}$, $j = 1, \ldots, N$.

A crucial property of LAD is its robustness against so-called y-outliers. This is a consequence of Theorem 1 which states that LAD is not affected by changes in y_i for data points that do not lie directly on the regression line as long as the signs of the residuals are not reverted.

Theorem 1 (Dodge 1997) *Suppose $\theta^\star$ is a minimizer of*

$$F(\theta) = \sum_{j=1}^{N} \left| y_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right) \right|.$$

Then, $\theta^\star$ is also a minimizer of

$$G(\theta) = \sum_{j=1}^{N} \left| z_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right) \right|,$$

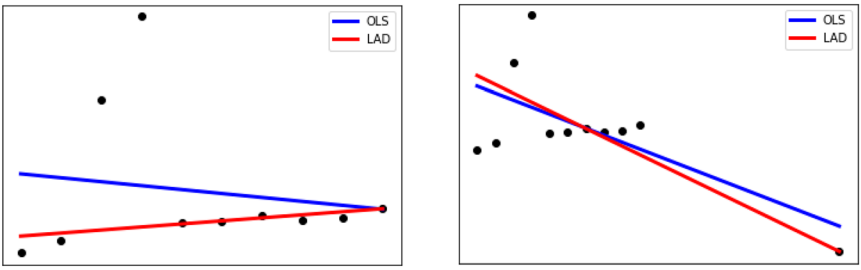
provided $z_j \geq \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j$ whenever $y_j > \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j$ and $z_j \leq \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j$ whenever $y_j < \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j$.

However, while being robust to y-outliers, LAD is still affected by *leverage points*, i.e., outliers in x. This is illustrated in Fig. 1.

Inspired by this observation, Rousseeuw (1984) proposed the *Least Trimmed Squares (LTS)* in 1984, whose formulation as an optimization problem is given by the *mixed-integer nonlinear optimization problem (MINLP)*

$$\min_{\theta, b} \sum_{j=1}^{N} r_{j,\theta}^2 \cdot b_j \qquad \text{s.t.} \sum_{j=1}^{N} b_j = N - k, \quad b \in \{0,1\}^N \tag{3}$$

with $k \in \mathbb{N}$ and $\frac{n}{2} < k < n$, where we use "$\cdot$" whenever multiplying decision variables. By design, (3) minimizes the sum of squares while ignoring the k largest squared deviations.



(a) In contrast to OLS, LAD is not affected by the presence of y-outliers

(b) OLS and LAD are both sensitive to leverage points

Fig. 1 Behavior of *ordinary least squares* (OLS) and *least absolute deviations* (LAD) in the presence of outliers of different types

Similar to the LTS, in 1999, the *least trimmed sum of absolute deviations (LTA)* is proposed by Hawkins and Olive (1999). LTA can be formulated as the MINLP

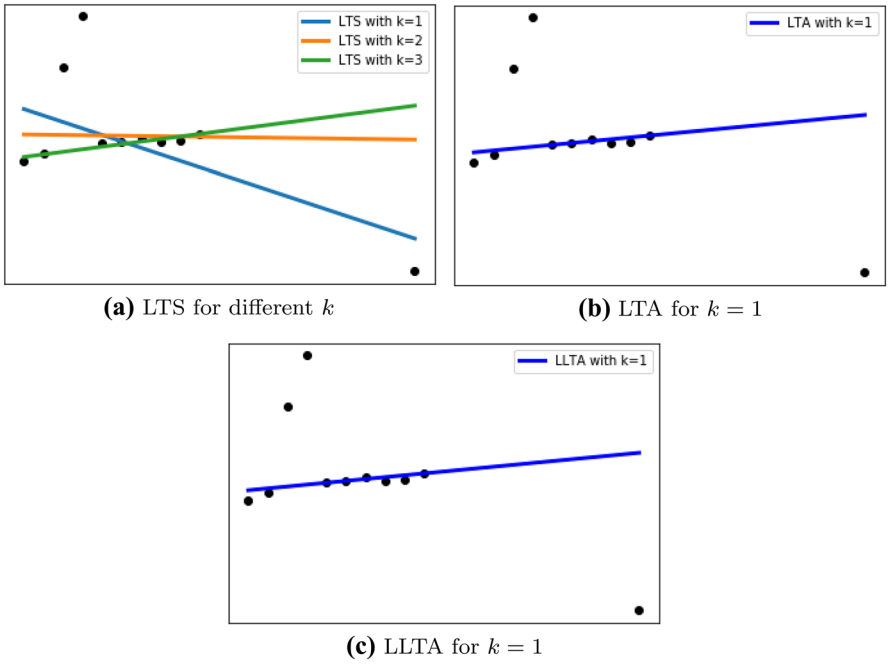
$$\min_{\theta,b} \sum_{j=1}^{N} |r_{j,\theta}| \cdot b_j \qquad \text{s.t.} \sum_{j=1}^{N} b_j = N - k, \quad b \in \{0,1\}^N. \tag{4}$$

To compute the LTA regression, Hawkins and Olive propose an enumeration algorithm over all possible subsets with k elements. This algorithm is of particular interest for the location model, as the LTA for a fixed subset is then obtained by evaluating the $N - k + 1$ subsets of ordered data points $y_{(k)}, y_{(k+1)}, \cdots, y_{(k+h-1)}$, for all $k = 1, \ldots, N - k + 1$ (Bassett 1991; Tableman 1994).

Next to an enumeration algorithm, the LTA regression problem is solved by Flores (2011) via a tailored continuous global optimization algorithm for the cases that the intercept is zero, i.e., $\theta_0 = 0$. First, the MINLP is reformulated as an NLP by introducing the nonconvex constraint $b_j^2 - b_j = 0$ for continuous variables b_j instead of the binary restriction on b_j. The resulting continuous nonconvex global optimization problem is then solved by a tailored global optimization algorithm in the spirit of Lasserre (2001).

The book about "open problems in optimization and data analysis" contains a chapter which discusses the connection between optimization and statistical robust estimators in the context of LTA regression (Pardalos and Migdalas 2018). For the location model, Zioutas et al. present a MINLP model of type (4) and a MILP reformulation of the bilinear terms using standard techniques. The LTA for the location model is extended to take into account outliers violating the correlational structure of the data set via a two-level approach in Chatzinakos et al. (2016).

Nowadays, within the statistics and machine learning communities, the LTS and LTA are solved by applying heuristics since both the LTS and the LTA are considered to be intractable as being $\mathcal{NP}$-hard (Bernholt 2006). However, during the period 1991-2015, due to algorithmic advances, *mixed-integer linear optimization problem (MILP)* solvers have experienced an average speedup factor of 780,000, *cf.* Bertsimas et al. 2016; Bixby 2012 and the references therein. These



(a) LTS for different k **(b)** LTA for $k = 1$

(c) LLTA for $k = 1$

Fig. 2 Behavior of LTS and LLTA in the presence of outliers of different types

machine-independent advances have been accompanied by an impressive progress in hardware performance. Thus, many real-world applications that could not be solved in the 1980s or 1990s are now solvable to global optimality within seconds. Similarly, modern software packages like CPLEX and GUROBI can now also solve large-scale nonconvex mixed-integer quadratic optimization problems.

Despite the latest solver developments, LTS and LTA can only be solved for medium-sized problem instances. Therefore, we introduce *Leveraged Least Trimmed Absolute Deviations (LLTA)*, which is a two-step approach that trains a linear function on possibly infiltrated data. The two steps are:

1. Identify all leverage points.
2. Minimize the total absolute deviations and ignore the $k \in \mathbb{N}$ largest deviations to data points *that are leverage points*, for some chosen k with $\frac{n}{2} < k < n$.

Consider now Fig. 2. LTS needs 11 binary decision variables and $k = 3$ to achieve a reasonable fit (Fig. 2a). LTA yields the same result with 11 binary decision variables and $k = 1$ (Fig. 2b). However, LLTA produces the same high-quality fit for $k = 1$ using only one binary decision variable (Fig. 2c).

These indicated advantages of LLTA compared to LTS and LTA are further examined in Sect. 3 after a formal introduction of LLTA in Sect. 2.

1.3 Statement of contributions

The unique contributions of this paper are: We

1. introduce *Leveraged Least Trimmed Absolute Deviations (LLTA)*,
2. demonstrate that LLTA outperforms LTS with respect to regression-quality and computational speed,
3. show that the regression-quality of LLTA is comparable to LTA while being much faster in terms of run time,
4. first benchmark the LTS and LTA with current MIQP solvers (the LTS and LTA are only solved by heuristic methods in the literature, ignoring the recent progress in MIQP algorithms and software developments).

The remainder of this paper is organized as follows. In Sect. 2, we introduce LLTA. We provide the benchmarking of LLTA with LTA and LTS in Sect. 3 before we conclude with Sect. 4.

2 Leveraged least trimmed absolute deviations

We start noting that the complexity of LTS and LTA is governed by

1. the number of ignored data points k since there are $\binom{N}{k}$ subsets of length k among the N data points and $\binom{N}{k}$ grows exponentially in k for fixed N and $k < N/2$,
2. and the number of binary variables since the search space also grows exponentially in the number of binary variables.

To mitigate the computational complexity resulting from the second point, we introduce *Leveraged Least Trimmed Absolute Deviations (LLTA)*. LLTA is a two-step procedure. Let $\mathcal{D} = \{1, \dots, N\}$.

1. Compute the index set $\mathcal{O} \subsetneq \mathcal{D}$ of leverage points; see Sect. 2.2 for details.
2. Solve the optimization problem

$$\min_{\theta, b} \sum_{j \in \mathcal{D} \setminus \mathcal{O}} |r_{j,\theta}| + \sum_{j \in \mathcal{O}} |r_{j,\theta}| \cdot b_j \qquad \text{s.t.} \sum_{j \in \mathcal{O}} b_j = |\mathcal{O}| - k, \quad b \in \{0,1\}^{|\mathcal{O}|}. \tag{5}$$

Since the classical ℓ_1-regression LAD is immune to y-outliers, we only protect our regression function with respect to leverage points. This is achieved through the parameter k allowing the optimal fit to ignore k data points within the index set of leverage points $\mathcal{O}$. Note that we utilize here that the set of leverage points

can be computed beforehand, while this is not possible for the y-outliers because they are regression-function dependent. Therefore, we obtain

1. a significant reduction of binary decision variables of (5) compared to (3) and (4), because LLTA only introduces one binary decision variable for each leverage point instead of one binary decision variable for each data point.
2. The possibility to choose smaller values of k compared to LTS, since LAD is already immune with respect to y-outliers.

Remark 1 As elaborated in Breiman (2001), there are two approaches to statistical modeling which are very different: In the *Data Modeling Culture*, a stochastic data model with an underlying distribution is assumed whose distribution is estimated from successive draws. Examples for that are given in Liu (1996) and Vanhatalo et al. (2009). In the *Algorithmic Modeling Culture*, a function $y = f(x)$ is fitted to observed data where the data generating process remains a black box with no further distributional assumptions. Many successful methods from machine learning like deep neural networks or gradient boosted trees are treated in the spirit of the latter culture. In this introductory work, we made the conscious decision to perform the analysis of LLTA within the framework of Algorithmic Modeling Culture. This yields a clear uncluttered overview about the main ideas and may serve as starting point for extensions from both cultures. To introduce underlying stochastic assumptions and to apply distributional-free sensitivity analysis using methods like bootstrapping are then possible extensions, *cf.* Sect. 2.5.

Remark 2 We assume that the number of infiltrated data points, i.e., the number of possible outliers, is strictly smaller than $N/2$. This is a standard assumption that is also posed in LTS and LTA. Therefore, also k must not exceed $N/2$ and we focus on the better half of the residuals as elaborated in Sect. 2.4.

2.1 Epigraph reformulation

In order to implement LTS, LTA and LLTA in a modern mixed-integer optimization solver, we first have to apply some reformulations. An LTS model is trained by solving the *mixed-integer quadratically-constraint quadratic optimization problem (MIQCQP)*

$$\begin{aligned} \min_{\theta,b,r} \ \frac{1}{N^2} \sum_{j\in\mathcal{D}}^{N} r_{j,\theta}^{\text{sqr}} \cdot b_j \quad \text{s.t.} \quad & r_{j,\theta}^{\text{sqr}} \geq \left(y_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right) \right)^2, \quad \forall j \in \mathcal{D} \\ & \sum_{j\in\mathcal{D}} b_j = N - k \\ & \theta \in \mathbb{R}^{n+1},\ r_\theta^{sqr} \in \mathbb{R}^N_{\geq 0},\ b \in \{0,1\}^N, \end{aligned}$$

where we have avoided the trilinear terms $r_{j,\theta}^2 \cdot b_j$ in the objective function by using only bilinear and quadratic expressions in the objective functions and constraints, respectively. Specifically, in an optimal solution,

$$r_{j,\theta}^{\text{sqr}} = \left(y_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right) \right)^2 = r_{j,\theta}^2.$$

An LTA-estimate is computed as an optimal solution of the *mixed-integer quadratic optimization problem (MIQP)*

$$\begin{aligned} \min_{\theta,b,r} \ \frac{1}{N} \sum_{j\in\mathcal{D}}^{N} r_{j,\theta}^{abs} \cdot b_j \quad \text{s.t.} \quad & r_{j,\theta}^{abs} \geq y_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right), \ \ \forall j \in \mathcal{D} \\ & r_{j,\theta}^{abs} \geq -y_j + \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j, \ \ \forall j \in \mathcal{D} \\ & \sum_{j\in\mathcal{D}} b_j = N - k \\ & \theta \in \mathbb{R}^{n+1}, \ r_\theta^{abs} \in \mathbb{R}^N_{\geq 0}, \ b \in \{0,1\}^N. \end{aligned}$$

The absolute value term $|r_{j,\theta}|$ in the objective function is modeled through two linear constraints, for every $j \in \mathcal{D}$. This is possible because LTS is a minimization problem.

Finally, we compute a linear regression function for LLTA by minimizing the MIQP

$$\begin{aligned} \min_{\theta,b,r} \ \frac{1}{N} \left(\sum_{j\in\mathcal{D}\setminus\mathcal{O}}^{N} r_{j,\theta}^{abs} + \sum_{j\in\mathcal{O}}^{N} r_{j,\theta}^{abs} \cdot b_j \right) \quad \text{s.t.} \quad & r_{j,\theta}^{abs} \geq y_j - \left(\theta_0 + \sum_{i=1}^{n} \theta_i x_i^j \right), \ \ \forall j \in \mathcal{D} \\ & r_{j,\theta}^{abs} \geq -y_j + \theta_0 + \sum_{i=1}^{n} \theta_i x_i^j, \ \ \forall j \in \mathcal{D} \\ & \sum_{j\in\mathcal{O}} b_j = |\mathcal{O}| - k \\ & \theta \in \mathbb{R}^{n+1}, \ r_\theta^{abs} \in \mathbb{R}^N_{\geq 0}, \ b \in \{0,1\}^{|\mathcal{O}|}, \end{aligned}$$

where we rewrite the absolute value terms like in the MIQP for the LTA above.

The prefactors $\frac{1}{N^2}$ and $\frac{1}{N}$ in the three formulations above do not affect the optimal solutions, but are added to enhance numerical stability. Because GUROBI version 9.0 is capable of dealing with MIQCQPs and MIQPs, there is no need to reformulate the optimization problems as MILPs. In this way, we avoid the introduction of Big-M constraints which are known to yield notoriously weak relaxations.

2.2 Computation of leverage points

Let $x^1, \ldots, x^j, \ldots, x^N \in \mathbb{R}^n$ be the data points and $q_i^{0.25}$ the *lower quartile* of their ith component. Further, let $q_i^{0.75}$ be the *upper quartile* and

$$\text{iqr}_i := q_i^{0.75} - q_i^{0.25}$$

the *interquartile range* of component $i \in \{1, \ldots, n\}$. Then, we introduce the following definition of a leverage point.

Definition 1 A data point $x \in \mathbb{R}^n$ is called *leverage point*, if for at least one $i \in \{1, \ldots, n\}$, $x_i < q_i^{0.25} - 1.5 \cdot \text{iqr}_i$ or $x_i > q_i^{0.75} + 1.5 \cdot \text{iqr}_i$.

This leads us to the definition of the index set of all leverage points

$$\mathcal{O} := \left\{ j \in \mathcal{D} \mid \exists i \in \{1, \ldots, n\} : x_i^j < q_i^{0.25} - 1.5 \cdot \text{iqr}_i \text{ or } x_i^j > q_i^{0.75} + 1.5 \cdot \text{iqr}_i \right\}.$$

The definition of a leverage point given in this paper is more specific than commonly defined in existing literature where a leverage point is usually described as "data point that has an extreme value for one of the explanatory variables" (Dodge 1997).

Remark 3 The outlier tolerance 1.5 might be treated as an hyper parameter t of LLTA which influences the number of binary variables as well as the prediction quality. However, for the remainder of this work, we set $t = 1.5$, which coincides with the definition of an outlier for boxplots (Tukey 1977).

Remark 4 Note that based on this definition of leverage points, there might be a tendency to identify more leverage points for high-dimensional data sets since the probability mass within a multidimensional probability distribution tends "to move away from its center" (van Handel 2014). The combination of domain knowledge and the selection of a tailored problem-specific outlier detection method (Hodge and Austin 2004) probably yields the best definition of leverage points for the problem at hand.

2.3 Choosing the number of outliers *k*

The choice of k may have a significant influence on the computed regression functions for LTS, LTA as well as LLTA. Quite generally, methods developed for LTS and LTA to choose k can also be applied toward LLTA.

By inspecting the optimization models (3), (4) and (5), we observe that their objective functions are monotone decreasing in the number k, i.e., allowing more outliers leads to a better fitting regression function. At the same time, increasing the number of outliers beyond the actual number of outliers in the data set implies loss of information. Consequently, k should not be chosen "too large." For practical problems, one would compute regression functions for different numbers of k and choose the k heuristically which seems to yield a good compromise between outlier

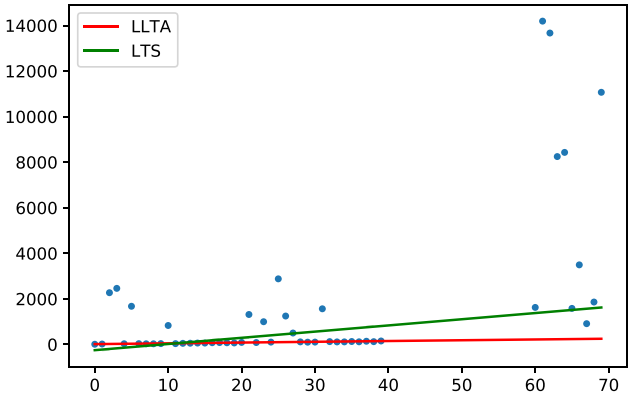


Fig. 3 Synthetic example with 50 data points, 20 of which are outliers, and regression lines computed by LLTA and LTS

detection and regression quality—one might choose k such that there is a significant improvement in the fit compared to $k-1$ and where $k+1$ yields only some minor improvement.

To choose k optimally, one would need an (objective) function quantifying both the regression fit and the "loss" from excluding potentially useful data.

2.4 Performance evaluation

Classical performance measures, like the root-mean-square error (RMSE) or mean-absolute error (MAE), are not suitable to measure the quality of statistical models in the presence of outliers because they evaluate the residuals for all data points. In contrast, a good robust model ignores some data points for being outliers. We evaluate the performance of the models by sorting the absolute residuals in ascending order, i.e.,

$$|r_{(1),\theta}| \leq \cdots \leq |r_{(j),\theta}| \leq |r_{(j+1),\theta}| \leq \cdots \leq |r_{(N),\theta}|$$

and computing the *trimmed MAE* on the better half of all residuals

$$\text{tMAE} := \frac{1}{\lfloor N/2 \rfloor} \sum_{j=1}^{\lfloor N/2 \rfloor} |r_{(j),\theta}|$$

where $\lfloor N/2 \rfloor$ denotes the floor function of $N/2$.

To motivate tMAE as performance metric, consider the following synthetic example where we have 30 "good" data points, ten "x-outliers," i.e., leverage points and ten "y-outliers." The scatter plot and regression lines for LLTA and LTS with $k = 10$ are depicted in Fig. 3.

It is not surprising to see that LTS is affected by the outliers, while LLTA yields a very good approximation of the "good" data points. However, how can we measure

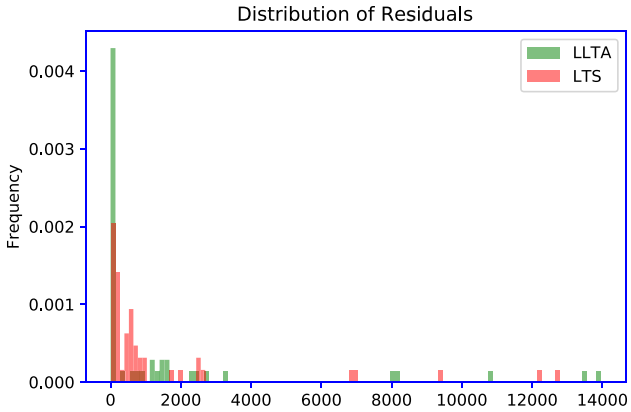


Fig. 4 Empirical distribution of residuals for LLTA and LTS

that in a multidimensional setting where visual inspections are not that easy to perform? We notice first that the classical performance measures RMSE and MAE are not suited since LTS outperforms LLTA in both, RMSE (3258 vs. 3660) and MAE (1471 vs. 1560), despite the fact that LTS' fit is obviously worse for this illustrative and synthetic example with 20 outliers. If we knew which of the data points are the "good" ones, we could just evaluate RMSE and MAE on these points. Unfortunately, we do not know that in real-world applications; otherwise, we would not have to immunize the regression function against outliers. Even worse, up to half of the data could be infiltrated and, in fact, we have an outlier rate of 40% in this example.

Let us take a look at the empirical distribution of the residuals of both methods which are illustrated in Fig. 4.

We recognize that the distributions are right-skewed—while the majority of all residuals are rather small, there exist some outliers, i.e., some data points show large residuals. This is not a problem per se since our goal is to design robust methods that ignore outliers on purpose. However, we should then also ignore these residuals when calculating our performance metric. Since this might affect up to 50% of our residuals, we decided to use trimmed MAE as performance metric for LLTA. If an upper bound b on the relative share of outliers is available, an obvious adjustment of tMAE would be to evaluate the residuals not only on the better half of the residuals but on $(1 - b) \cdot 100\%$ of the data.

2.5 Uncertainty measurement

The quantification of prediction uncertainty is important as it tries to determine the trustworthiness of a particular prediction or even of the statistical model in general. In particular, if decision making is based on good predictions, an uncertainty measure for the prediction is crucial.

A prominent example for model predictions, embedded in a "predict-tell" cycle, is Bayesian optimization. In Bayesian optimization, in each iteration an acquisition function is optimized, taking into account the prediction of a surrogate model

as well as the model uncertainty (Pelikan et al. 1999). The predominant surrogate models are Gaussian Processes which rely on a normal distribution assumption also yielding an uncertainty estimate.

However, more recent distributional-free approaches to Bayesian optimization also work with gradient tree ensemble methods as surrogate models and distance-based uncertainty measures (Thebelt et al. 2020a). A distance-based uncertainty measure does not assume any probability distribution in the data and is therefore also applicable to LLTA. The main idea is to measure the distance of an x-value, whose y-value is to be predicted, to the set of existing training data since the statistical model might have bad extrapolation properties. Distance-based uncertainty measures have a nice intuitive interpretation, but might provide misleading information for large datasets where especially the ℓ_2-norm shows counterintuitive behavior (Aggarwal et al. 2001). Distance-based uncertainty measures that use the ℓ_1-norm might be a useful uncertainty measure for LLTA.

Next to distance-based uncertainty measures, the second (by far more popular) approach to uncertainty estimation without distributional assumptions is bootstrapping (Diaconis and Efron 1983). Koenker and Hallock use bootstrapping in their famous work (Koenker and Hallock 2001) to quantify uncertainty in quantile regression which is a very popular approach to robust regression. The main idea of bootstrapping is to train statistical models on random samples of the training data and to compare its statistical properties. If models trained on different samples tend to vary much, then this might be an indication of a high model uncertainty. Therefore, using bootstrapping, or one of its many variants, is recommended as uncertainty measure for LLTA.

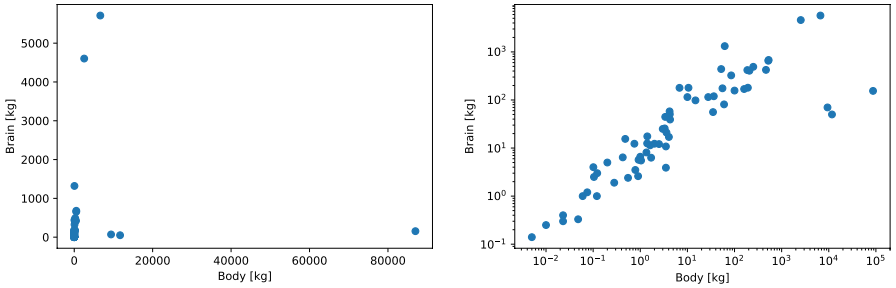
3 Computational results

We perform computational tests, comparing the two approaches LTS and LTA from the literature to the new model LLTA. All models are implemented in GUROBI 9.0 via its Python-API, and they are solved on a standard desktop computer possessing four cores each with 2.71 GHz and 16 GB RAM.

We start with an example on the body–brain data set in Sect. 3.1 with the goal to illustrate the usage of LLTA and to give an indication regarding the strengths and weaknesses of the examined models. We then use 11 instances from the literature in Sect. 3.2 to demonstrate the computational differences of the models resulting from LTS, LTA and LLTA.

3.1 Comparison with LTS and LTA on the body–brain data set

To illustrate the new model LLTA, we compare it to LTS and LTA based on the "Brain and Body Weights" dataset (Rousseeuw and Leroy 1987; Weisberg 1985). This dataset contains the following average body and brain weights in kg in the format (body, brain) for 65 animals



(a) Unscaled **(b)** Log-scaled

Fig. 5 Pairs of body–brain weights for 65 species

(1.35, 8.1), (465, 423), (36.33, 119.5), (27.66, 115), (1.04, 5.5), (11700, 50), (2547, 4603), (187.1, 419), (521, 655), (10, 115), (3.3, 25.6), (529, 680), (207, 406), (62, 1320), (6654, 5712), (9400, 70), (6.8, 179), (35, 56), (0.12, 1), (0.023, 0.4), (2.5, 12.1), (55.5, 175), (100, 157), (52.16, 440), (0.28, 1.9), (87000, 154.5), (0.122, 3), (192, 180), (3.385, 44.5), (0.48, 15.5), (14.83, 98.2), (4.19, 58), (0.425, 6.4), (0.101, 4), (0.92, 5.7), (1, 6.6), (0.005, 0.14), (0.06, 1), (3.5, 10.8), (2, 12.3), (1.7, 6.3), (0.023, 0.3), (0.785, 3.5), (0.2, 5), (1.41, 17.5), (85, 325), (0.75, 12.3), (3.5, 3.9), (4.05, 17), (0.01, 0.25), (1.4, 12.5), (250, 490), (10.55, 179.5), (0.55, 2.4), (60, 81), (3.6, 21), (4.288, 39.2), (0.075, 1.2), (0.048, 0.33), (3, 25), (160, 169), (0.9, 2.6), (1.62, 11.4), (0.104, 2.5), (4.235, 50.4)

which are depicted in Fig. 5a, b using logarithmic scales.

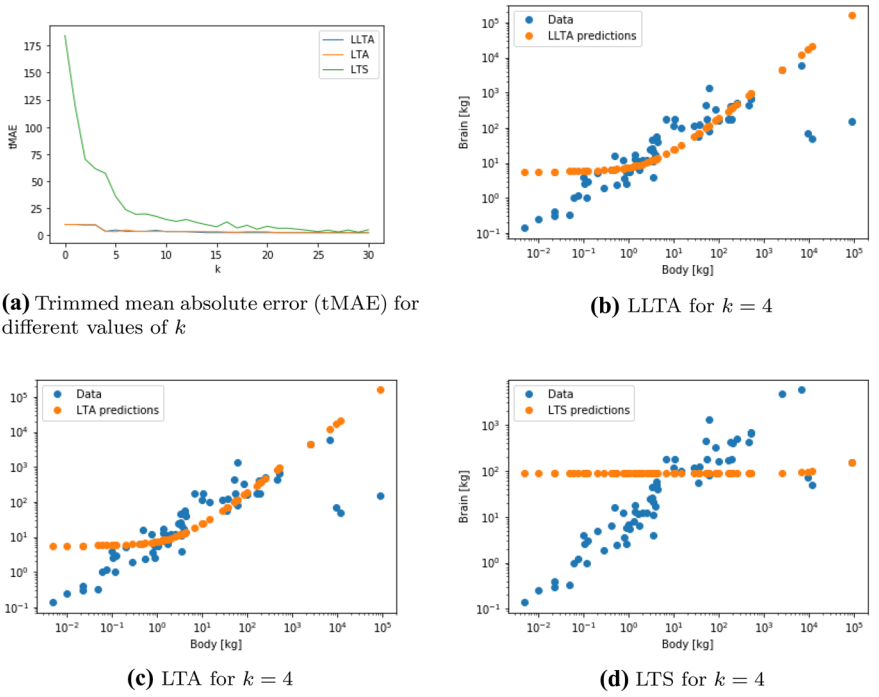
For the body–brain data set, we have $n = 1$. Its quartiles are given by $q^{0.25} = 0.75$ and $q^{0.75} = 60$ which results in an interquartile range of iqr $= 59.25$. We compute

$$\begin{aligned}\mathcal{O} &= \{j \in \mathcal{D} |\, x_j^1 < 0.75 - 1.5 \cdot 59.25 \text{ or } x_j^1 > 60 + 1.5 \cdot 59.25\} \\ &= \{j \in \mathcal{D} |\, x_j^1 > 148.875\} \\ &= \{2, 6, 7, 8, 9, 12, 13, 15, 16, 26, 28, 52, 61\},\end{aligned}$$

i.e., we have 13 leverage points with the x-values 465, 11700, 2547, 187.1, 521, 529, 207, 6654, 9400, 87000, 192, 250, 160.

Due to the presence of these leverage points, OLS and LAD are heavily affected such that there is need for a robust statistical model. Therefore, we train LTS, LTA and LLTA on the data set for different values of k. In Fig. 6, we observe that the asymptotic trimmed mean absolute errors tMAE of the residuals of the three models is comparable, whereas LTA and LLTA obtain a better score for $k < 25$.

In addition to the good statistical performance of LLTA, we observe in Fig. 7 that LLTA outperforms LTS and LTA with respect to the number of visited nodes in the branch-and-bound tree and with respect to the run time (at a time limit of 600 s). LLTA visits at most 151 nodes and solves most optimization problems within the root node. In contrast, LTS and LTA visit up to 155,572 and 1,071,225 nodes, respectively. Regarding the run time, LLTA needs at most 0.04 s to solve



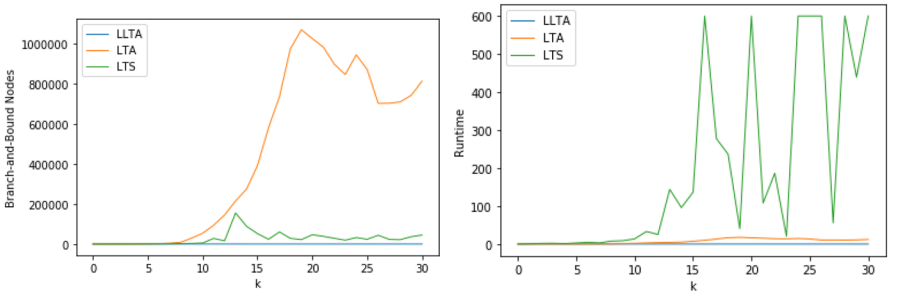
(a) Trimmed mean absolute error (tMAE) for different values of k

(b) LLTA for $k = 4$

(c) LTA for $k = 4$

(d) LTS for $k = 4$

Fig. 6 Trimmed mean absolute errors and model instances of LLTA, LTA and LTS for different values of k on log-scaled axes



(a) Branch-and-Bound nodes

(b) Run time

Fig. 7 Number of visited branch-and-bound nodes and run times for LLTA, LTS and LTA with a time limit of 120 s for different values of k

any of the instances in contrast to LTA whose run time increases up to 17 seconds. In turn, LTA is much better than LTS, where an optimality certificate cannot be computed within the time limit. As such, we obtain a maximum speedup of 425 (or 99.8%) of LLTA compared to LTA and of 15,000 (or 99.99%) of LLTA to LTS.

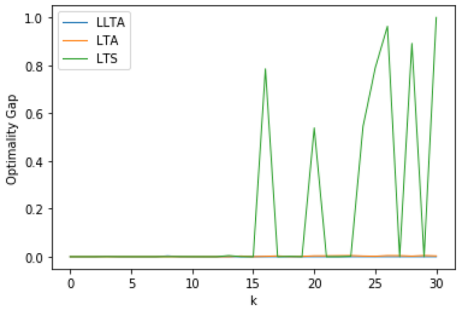


Fig. 8 Remaining optimality gap with time limit of 600 s

Table 1 Some information on the datasets where rows refer to the number of observations and columns are the number of features

#	Title	Rows	Columns	Source
1	Coleman data set	20	6	Rousseeuw and Leroy (1987)
2	Delivery time data	25	3	Montgomery and Peck (1982)
3	Hawkins, Bradu, Kass's Artificial Data	75	4	Hawkins et al. (1984)
4	Heart catherization data	12	3	Rousseeuw and Leroy (1987), Weisberg (1985)
5	Waterflow measurements of Kootenay	13	2	Ezekiel and Fox (1959)
6	Pension funds data	18	2	Rousseeuw and Leroy (1987)
7	Phosphorus content data	18	3	Rousseeuw and Leroy (1987)
8	Salinity data	28	4	Ruppert and Carroll (1980)
9	Siegel's exact fit example data	9	2	Rousseeuw and Leroy (1987)
10	Steam usage data (excerpt)	25	9	Norman and Draper (1981)
11	Modified data on wood specific gravity	20	6	Rousseeuw and Leroy (1987)

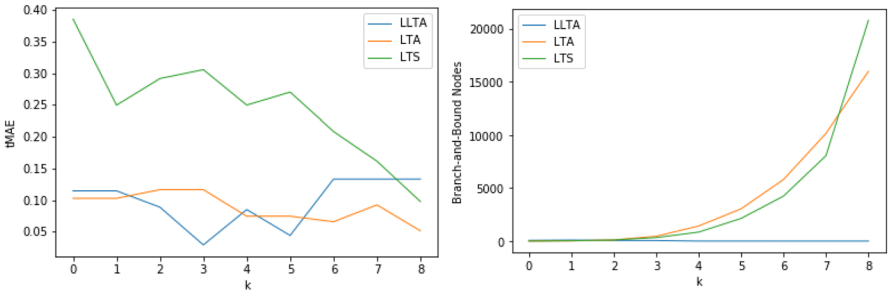
For some values $k \geq 16$, LTS does not manage to close the optimality gap within the time limit, as depicted in Fig. 8.

3.2 Datasets from the literature

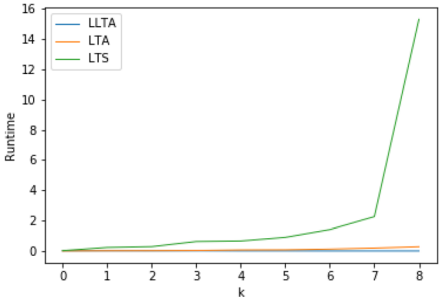
We extracted 11 datasets from the existing literature on robust regression. Table 1 summarizes some relevant information about the datasets we use for benchmarking.

We run LTS, LTA and LLTA for all data sets at a time limit of 600 seconds for all $k \in \{0, \ldots, \lfloor N/2 - 1 \rfloor\}$ for LTS and LTA as well as $k \in \{0, \ldots, |\mathcal{O}|\}$ for LLTA. The results are depicted in Figs. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19, where we see the trimmed MAE, the number of visited branch-and-bound nodes and the run time for each method. Among the 11 datasets, we compare 118 different regression functions.[1]

[1] There are nine regression functions for data set # 1, 11 (# 2), 36 (# 3), 5 (# 4), 5 (# 5), 8 (# 6), 8 (# 7), 13 (# 8), 3 (# 9), 11 (# 10), 9 (# 11)



(a) tMAE **(b)** Branch-and-Bound nodes

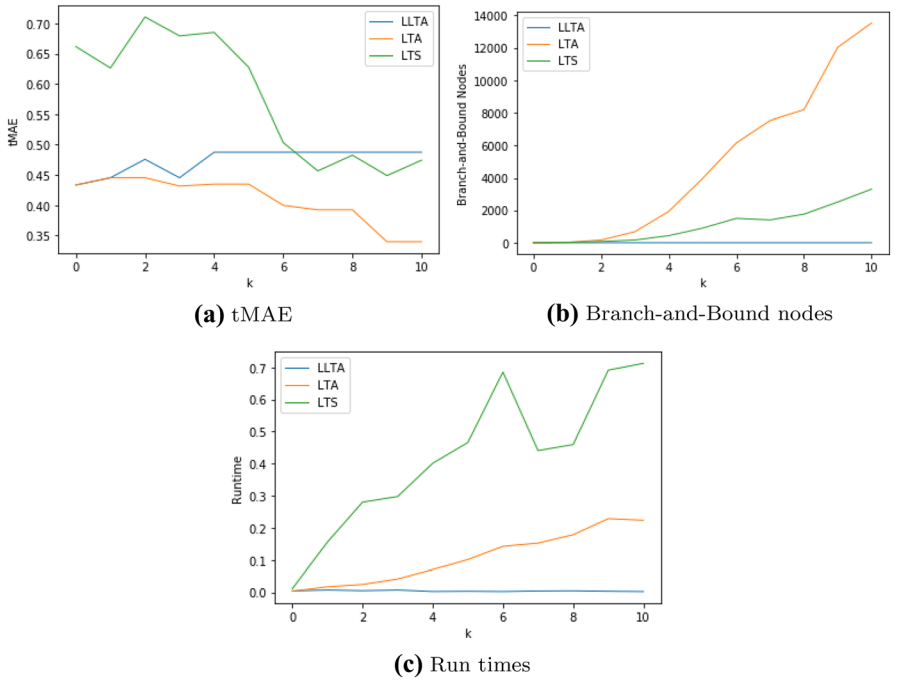


(c) Run times

Fig. 9 Coleman data set

Consider now Figs. 9a, 10a, 11a, 12a, 13a, 14a, 15a, 16a, 17a, 18a and 19a, where the tMAE is shown for different values of k. Among the 118 regression functions, the performance of LTS is never better than the one of LTA. This is because LTS is not immune against y-outliers and, thus, requires larger values of k to achieve a similar performance than LTA. For 68 instances, LLTA performs better than LTS and for $k < 4$, LLTA is always better. Note that these results are heavily affected by dataset # 3 (Fig. 11a). LLTA performs comparable to LTA in most datasets. Exceptions are datasets # 2 (Fig. 10a), # 7 (Fig. 15a) and # 10 (Fig. 18a), where LTA is consistently better than LLTA and dataset # 9 (Fig. 17a), where LLTA outperforms LTA. It is remarkable that both LTA and LLTA have a quite similar performance, given that LTA has more degrees of freedom (because it can choose the leverage points among all data points compared to LLTA which is restricted to the index set of leverage points $\mathcal{O}$). Even more surprising is that LLTA shows a (strictly) better tMAE compared to LTA for 12[2] regression functions! Note that the computed regression functions are evaluated with respect to the tMAE (measuring the better half of the

[2] # 1 (Fig. 9a) $K = 2,3,5$; # 3 (Fig. 11a) $K = 6,12$; # 4 (Fig. 12a) $K = 2$; #8 (Fig. 16a) $K = 2,3$; # 9 (Fig. 17a) $K = 1,2$; #11 K = 3,4



(a) tMAE **(b)** Branch-and-Bound nodes

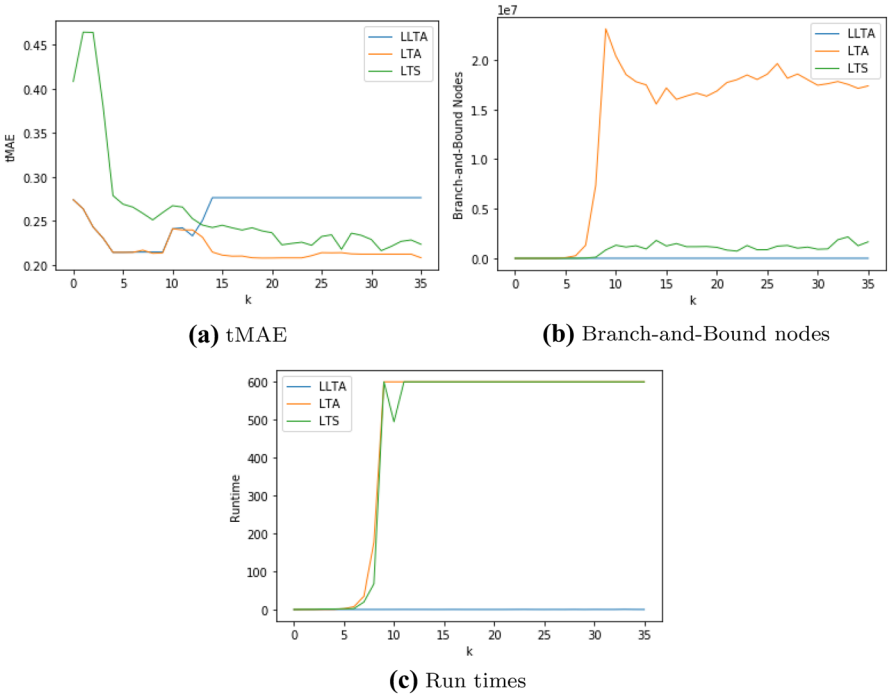
(c) Run times

Fig. 10 Delivery time data

residuals) and not with respect to the objective function (measuring all residuals except the k outliers). The difference in evaluation metrics and objective function also explains why the tMAE curves are not monotone decreasing in k (while we still observe a decreasing trend with an increase in k). This nonmonotone behavior can be seen, for example, in dataset # 1 (Fig. 9a).

Figures 9b, 10b, 11b, 12b, 13b, 14b, 15b, 16b, 17b, 18b and 19b show the number of visited branch-and-bound nodes, necessary to solve the corresponding instances. The trend here is clear: While LLTA shows a linear growth in k, both LTA and LTS follow an exponential curve. This behavior is by design, as the primary motivation to introduce LLTA is the significant reduction of the computational burden. In many instances, the LLTA can be solved in the root node.

The computational time is highly related to the number of visited branch-and-bound nodes. Therefore, Figs. 9c, 10c, 11c, 12c, 13c, 14c, 15c, 16c, 17c, 18c and 19c show a similar trend than the number of visited branch-and-bound nodes. In addition, we observe that the computational efforts to solve LTS are significantly greater than solving LTA. A similar percentage increase in runtime is observable for LTA compared to LLTA. Note that LLTA can solve any instance in at most 0.66 s, except the one instance for dataset #10 and $k = 0$ (Fig. 18c). In average, LLTA is 699.58 faster compared to LTA and 797.08 faster compared to LTS. However, the average speedup is mainly driven by dataset #3 where LLTA is 7543 faster than LTA



(a) tMAE **(b)** Branch-and-Bound nodes

(c) Run times

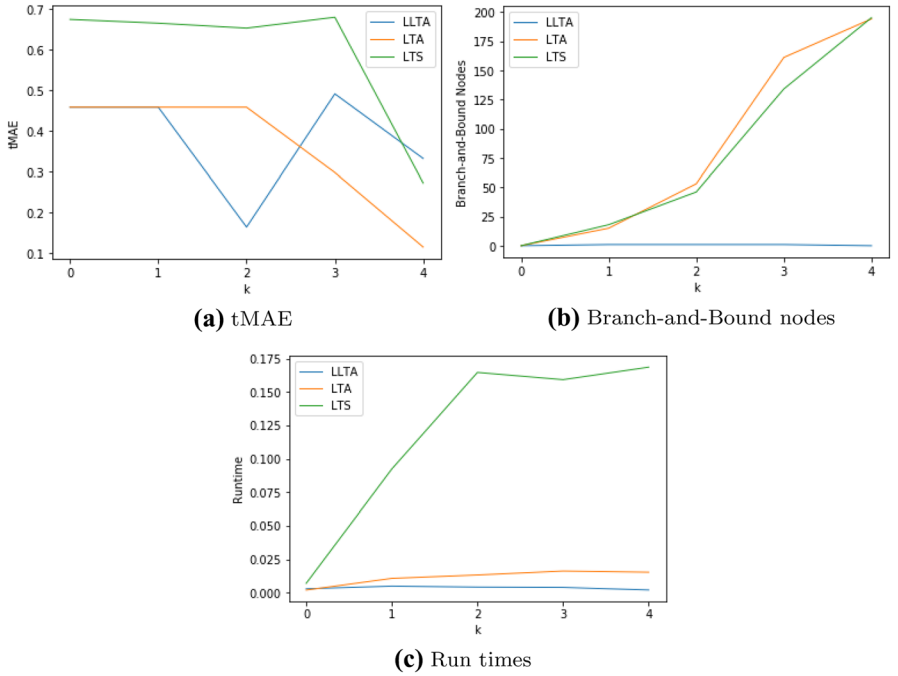
Fig. 11 Hawkins, Bradu, Kass's artificial data

and 7436 faster than LTS. Then, median speedup values are 5.33 compared to LTA and 80.23 compared to LTS.

Instance # 3 (Fig. 11a) is of particular interest due to its size. With 75 rows, this dataset contains about three times more rows than any other dataset (cf. Table 1). When inspecting Fig. 11b, c, we see that for $k \geq 11$, none of the instances for LTA and LTS can be solved to optimality within the time limit. This explains why we do not observe an exponential grows in the number of visited branch-and-bound nodes for large k for this dataset.

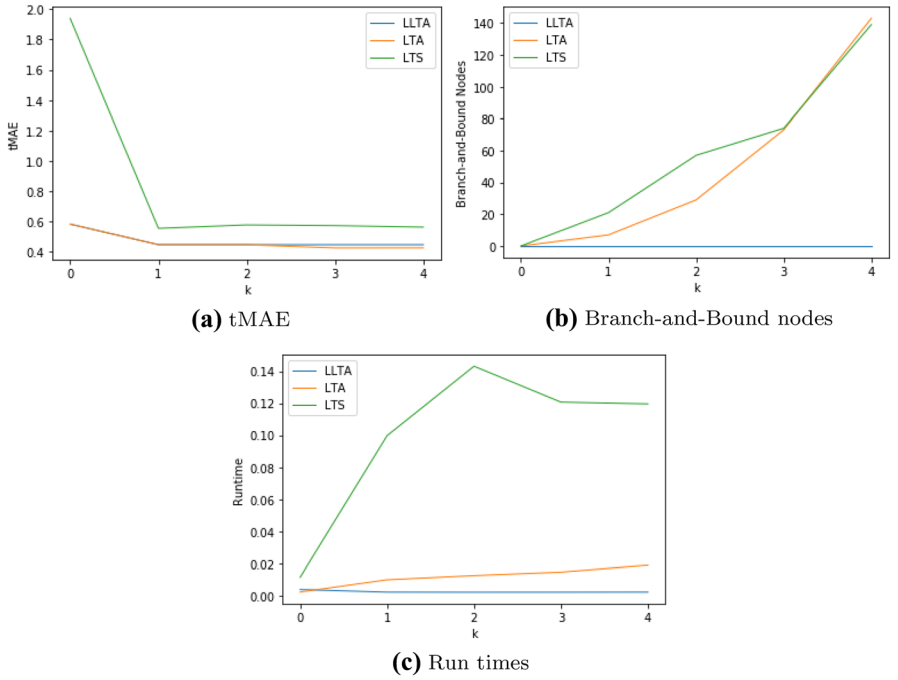
4 Summary and outlook

We introduce the Leveraged Least Trimmed Absolute Deviations (LLTA) which is based on the Least Trimmed Absolute Deviations (LTA). We make use of two observations. First, LTA is by design immune against y-outliers. Second, the leverage points can be computed beforehand in contrast to the y-outliers, because the y-outliers depend on the constructed regression function. As such, LLTA combines the advantages of LTA while considering only leverage points as potential x-outliers. This has the consequence that the proposed regression model LLTA is immune against both leverage points and y-outliers. At the same time, the computational burden is much lower compared to LTA.



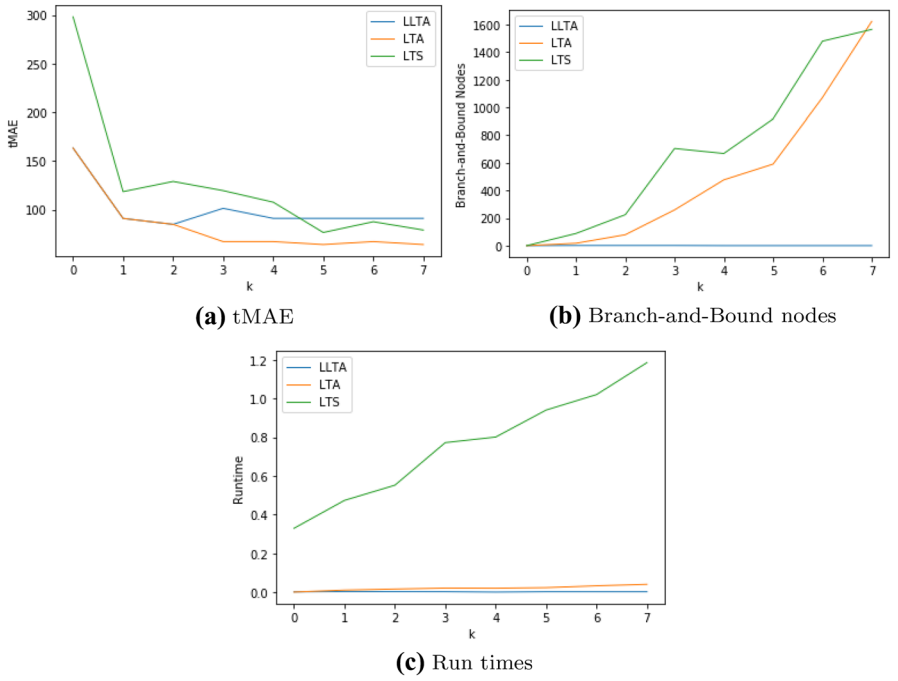
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 12 Heart catherization data



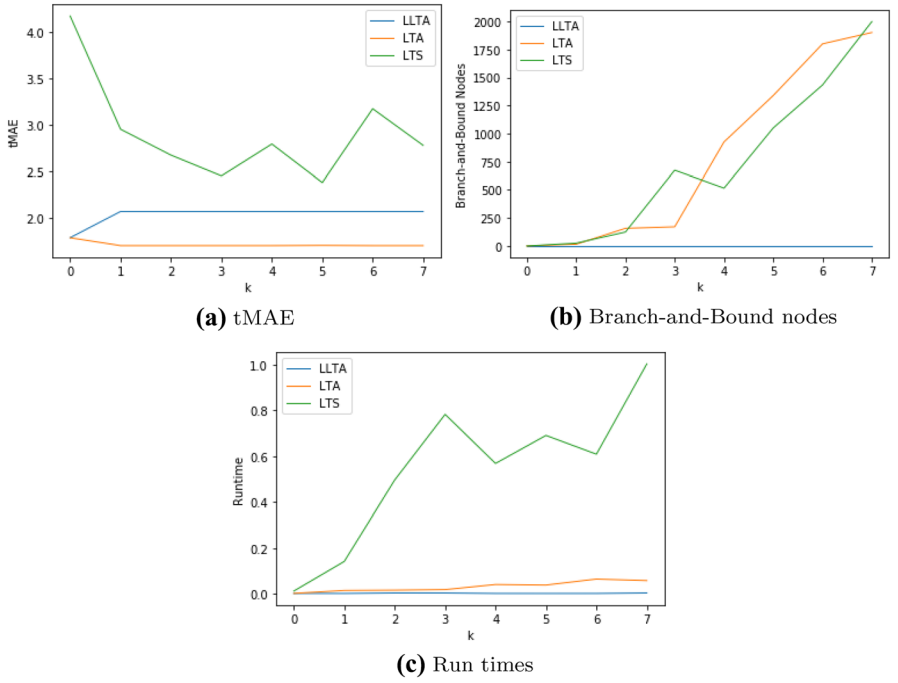
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 13 Waterflow measurements of Kootenay River in Libby and Newgate



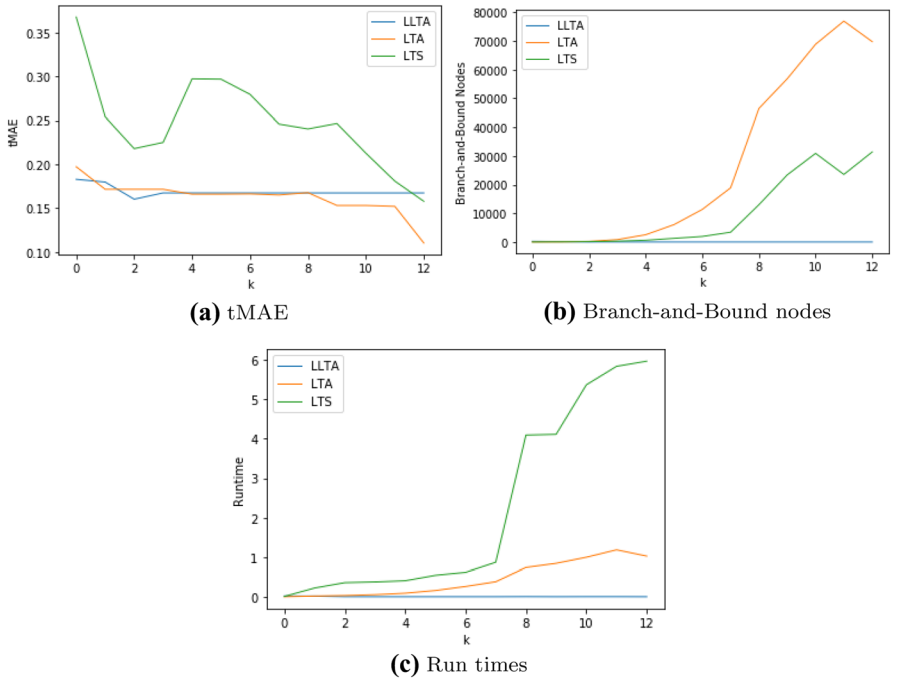
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 14 Pension funds data



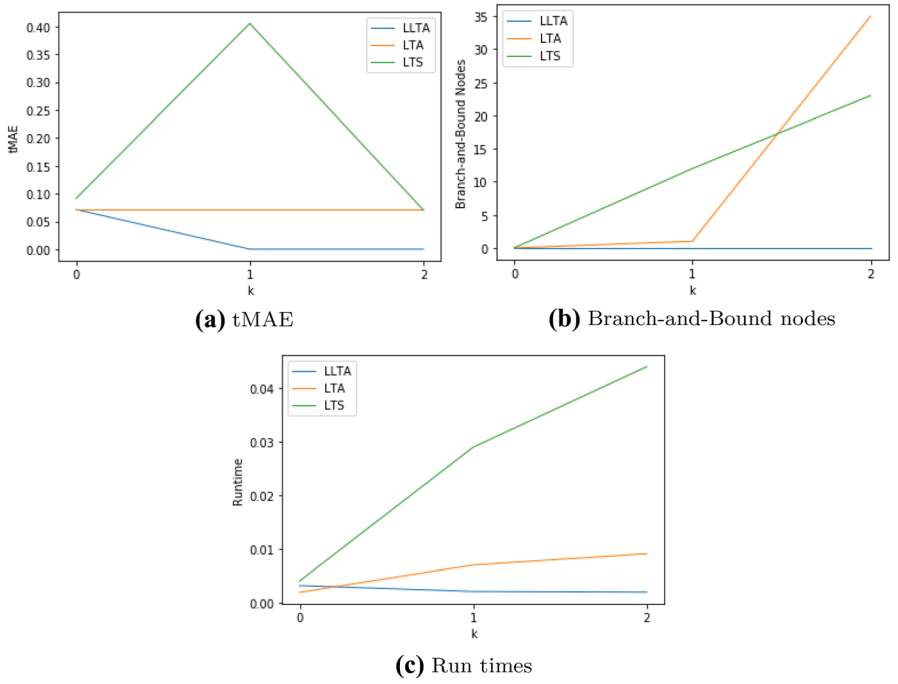
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 15 Phosphorus content data



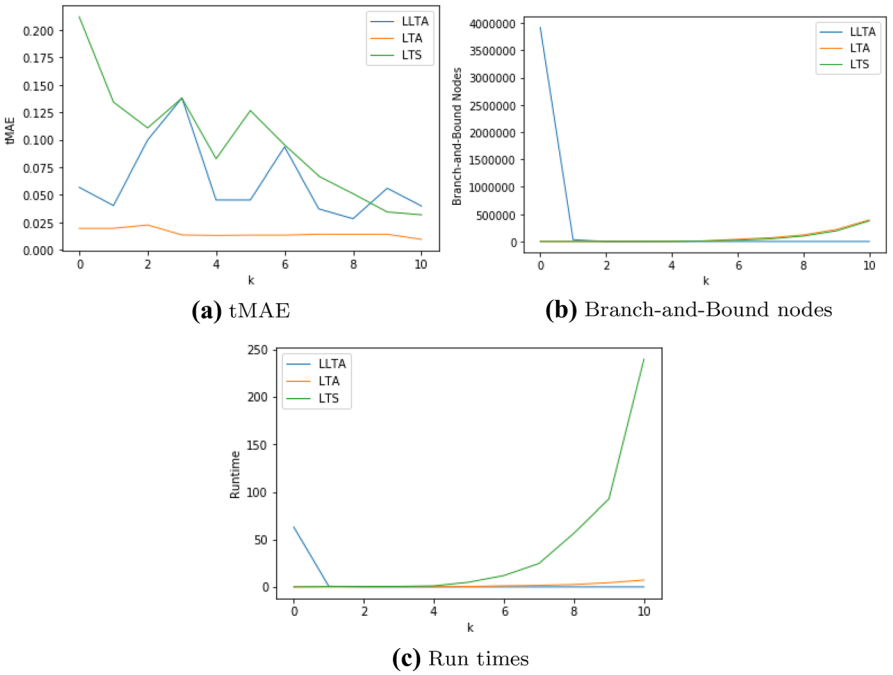
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 16 Salinity data



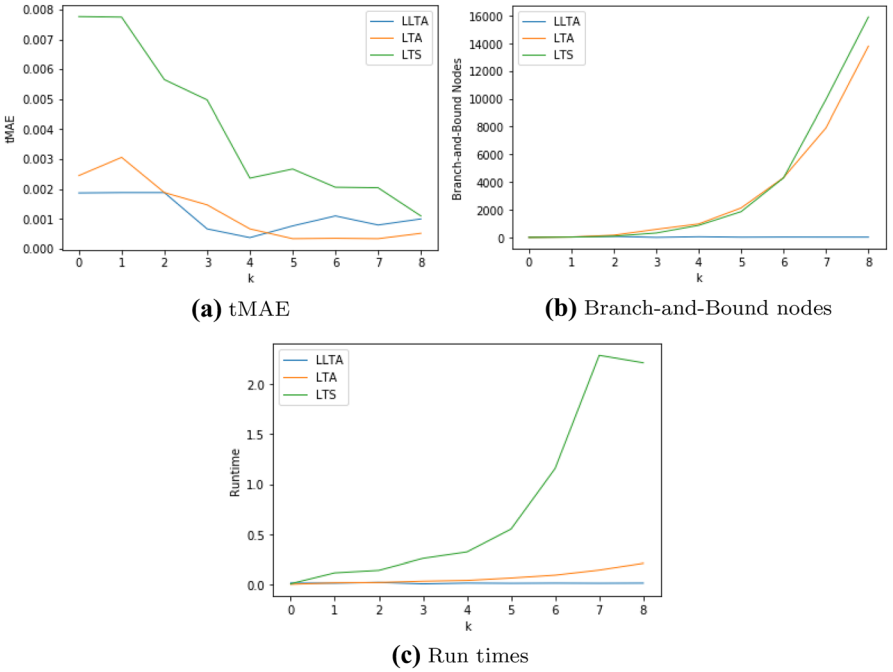
(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 17 Siegel's exact fit example data



(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 18 Steam usage data (excerpt)



(a) tMAE **(b)** Branch-and-Bound nodes **(c)** Run times

Fig. 19 Modified data on wood specific gravity

Our computational results on known benchmark instances show that LLTA has a comparable performance of the computed regression models compared to LTA. At the same time, LLTA can be solved much faster compared to LTA. The computed regression models by LLTA tend to outperform the ones computed by the well-known least trimmed squares (LTS). For small k, this effect is drastic. In addition, LLTA can be solved several orders of magnitude faster than LTS.

Acknowledgements We thank Peter Rousseuw for giving a presentation at BASF about robust statistics. This and an interesting discussion with Jakob Raymaekers and Robert Matthew Lee inspired the creation of this article.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche J, Vianu V (eds) Database theory—ICDT 2001. Springer, Berlin, pp 420–434

Bassett GW Jr (1991) Equivariant, monotonic, 50% breakdown estimators. Am Stat 45(2):135–137

Bernholt T (2006) Robust estimators are hard to compute. Tech. rep

Bertsimas D, Dunn J (2019) Machine learning under a modern optimization lens. Dynamic Ideas LLC. https://books.google.de/books?id=g3ZWygEACAAJ

Bertsimas D, King A, Mazumder R (2016) Best subset selection via a modern optimization lens. Ann Stat 44:813–852

Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin

Bixby RE (2012) A brief history of linear and mixed-integer programming computation. Doc Math Extra vol.: Optimization Stories:107–121

Breiman L (2001) Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat Sci 16(3):199–231

Chatzinakos C, Pitsoulis L, Zioutas G (2016) Optimization techniques for robust multivariate location and scatter estimation. J Comb Optim 31(4):1443–1460

Clark LA, Pregibon D (2017) Tree-based models. In: Statistical models in S. Routledge, pp 377–419

Diaconis P, Efron B (1983) Computer-intensive methods in statistics. Sci Am 248(5):116–131

Dodge Y (1997) Lad regression for detecting outliers in response and explanatory variables. J Multivar Anal 61(1):144–158

Ezekiel M, Fox KA (1959) Methods of correlation and regression analysis: linear and curvilinear. Wiley, Hoboken

Flores S (2011) Global optimization problems in robust statistics. Ph.D. thesis
Goldberg N, Rebennack S, Kim Y, Krasko V, Leyffer S (2021) MINLP formulations for continuous piecewise linear function fitting. Comput Optim Appl
Hastie T, Tibshirani R, Tibshirani RJ (2017) Extended comparisons of best subset selection, forward stepwise selection, and the lasso. ArXiv preprint arXiv:1707.08692
Hawkins DM, Olive D (1999) Applications and algorithms for least trimmed sum of absolute deviations regression. Comput Stat Data Anal 32(2):119–134
Hawkins DM, Bradu D, Kass GV (1984) Location of several outliers in multiple-regression data using elemental sets. Technometrics 26(3):197–208
Hodge V, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22(2):85–126
Jünger M, Liebling TM, Naddef D, Nemhauser GL, Pulleyblank WR, Reinelt G, Rinaldi G, Wolsey LA (2009) 50 Years of integer programming 1958–2008: from the early years to the state-of-the-art. Springer, Berlin
Koenker R, Bassett G et al (1985) On Boscovich's estimator. Ann Stat 13(4):1625–1628
Koenker R, Hallock KF (2001) Quantile regression. J Econ Perspect 15(4):143–156
Krasko V, Rebennack S (2017) Two-stage stochastic mixed-integer nonlinear programming model for post-wildfire debris flow hazard management: Mitigation and emergency evacuation. Eur J Oper Res 263(1):265–282
Lasserre JB (2001) Global optimization with polynomials and the problem of moments. SIAM J Optim 11(3):796–817
Liu C (1996) Bayesian robust multivariate linear regression with incomplete data. J Am Stat Assoc 91(435):1219–1227
Schmidt M, Le Roux N, Bach F (2017) Minimizing finite sums with the stochastic average gradient. Math Program 162:83–112
Marsland S (2015) Machine learning: an algorithmic perspective. CRC Press, Boca Raton
Micula G, Micula S (2012) Handbook of splines, vol 462. Springer, Berlin
Montgomery DC, Peck EA (1982) Introduction to linear regression analysis. Wiley, Hoboken
Norman R, Draper HS (1981) Applied regression analysis, 2nd edn. Wiley, Hoboken
Pardalos PM, Migdalas A (2018) Open problems in optimization and data analysis, vol 141. Springer, Berlin
Pelikan M, Goldberg DE, Cantú-Paz E et al (1999) Boa: the bayesian optimization algorithm. In: Proceedings of the genetic and evolutionary computation conference GECCO-99, vol 1. Citeseer, pp 525–532
Rebennack S, Kallrath J (2015) Continuous piecewise linear delta-approximations for univariate functions: computing minimal breakpoint systems. J Optim Theory Appl 167(2):617–643
Rebennack S, Krasko V (2020) Piecewise linear function fitting via mixed-integer linear programming. INFORMS J Comput 32(2):507–530
Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22(3):400–407
Rousseeuw PJ (1984) Least median of squares regression. J Am Stat Assoc 79(388):871–880
Rousseeuw PJ, Leroy AM (1987) Robust regression and outlier detection. Wiley, Hoboken
Ruppert D, Carroll RJ (1980) Trimmed least squares estimation in the linear model. J Am Stat Assoc 75(372):828–838
Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959
Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2(6):568–576
Stigler SM (1981) Gauss and the invention of least squares. Ann Stat 9(3):465–474
Tableman M (1994) The asymptotics of the least trimmed absolute deviations (LTAD) estimator. Stat Probab Lett 19(5):387–398
Thebelt A, Kronqvist J, Lee RM, Sudermann-Merx N, Misener R (2020a) Global optimization with ensemble machine learning models. In: Pierucci S, Manenti F, Bozzano GL, Manca D (eds) 30th European Symposium on computer aided process engineering, computer aided chemical engineering, vol 48. Elsevier, Amsterdam, pp 1981–1986. https://doi.org/10.1016/B978-0-12-823377-1.50331-1
Thebelt A, Kronqvist J, Mistry M, Lee RM, Sudermann-Merx N, Misener R (2020b) ENTMOOT: a framework for optimization over ensemble tree models. arXiv:2003.04774
Tukey JW (1977) Exploratory data analysis, vol 2. Addison-Wesley, Reading
van Handel R (2014) Probability in high dimension. Tech. rep., Princeton University NJ

Vanhatalo J, Jylänki P, Vehtari A (2009) Gaussian process regression with student-t likelihood. In: Bengio Y, Schuurmans D, Lafferty JD, Williams CKI, Culotta A (eds.) Advances in Neural Information Processing Systems 22, pp. 1910–1918. Curran Associates, Inc. http://papers.nips.cc/paper/3806-gaussian-process-regression-with-student-t-likelihood.pdf

Weisberg S (1985) Applied linear regression, 2nd edn. Wiley, Hoboken

Wolsey LA, Nemhauser GL (1999) Integer and combinatorial optimization, vol 55. Wiley, Hoboken