Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 52 (2007) 763–772
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

Computing the least quartile difference estimator in the plane☆

Thorsten Bernholt[a], Robin Nunkesser[a,*], Karen Schettlinger[b]

[a]FB Informatik, LS 2, Universität Dortmund, 44221 Dortmund, Germany
[b]FB Statistik, Universität Dortmund, 44221 Dortmund, Germany

Available online 2 January 2007

Abstract

A common problem in linear regression is that largely aberrant values can strongly influence the results. The least quartile difference (LQD) regression estimator is highly robust, since it can resist up to almost 50% largely deviant data values without becoming extremely biased. Additionally, it shows good behavior on Gaussian data—in contrast to many other robust regression methods. However, the LQD is not widely used yet due to the high computational effort needed when using common algorithms. It is shown that it is possible to compute the LQD estimator for n bivariate data points in expected running time $\mathcal{O}(n^2 \log n)$ or deterministic running time $\mathcal{O}(n^2 \log^2 n)$. Additionally, two easy to implement algorithms with slightly inferior time bounds are presented. All of these algorithms are also applicable to least quantile of squares and least median of squares regression through the origin, improving the known time bounds to expected time $\mathcal{O}(n \log n)$ and deterministic time $\mathcal{O}(n \log^2 n)$. The proposed algorithms improve on known results of existing LQD algorithms and hence increase the practical relevance of the LQD estimator.

Keywords: Computational statistics; Computational geometry; Robust regression; Least quartile difference (LQD); Least median of squares (LMS); Least quantile of squares (LQS)

1. Introduction

Least squares (LS) is one of the most popular regression methods since it is computationally simple and it has minimal variance for Gaussian distributed data. However, the LS estimator can be strongly influenced by outlying values. The aim of robust regression in the plane is to fit a straight line through a set of two-dimensional points in such a way that outliers do not affect the fit.

To quantify the robustness of an estimator, Donoho and Huber (1983) define the (*finite sample*) *breakdown point* as the smallest fraction of data points that needs to be changed to have an unbounded effect on the estimate. Thus here, the term 'robust' stands for a high breakdown value. The LS estimator is not robust, as its breakdown value is $1/n$, i.e. a single outlier can have arbitrarily large effects on the estimation.

The *least quartile difference* (LQD) estimator, introduced by Croux et al. (1994), has a breakdown point of $\lfloor n/2 \rfloor / n$ if the data fulfill certain requirements. This means that up to 50% of the data can be contaminated without ruining the

☆ The financial support of the Deutsche Forschungsgemeinschaft (SFB 475, Reduction of complexity in multivariate data structures) is gratefully acknowledged.

* Corresponding author. Tel.: +49 231 755 5132; fax: +49 231 755 2047.
E-mail addresses: thorsten.bernholt@udo.edu (T. Bernholt), robin.nunkesser@udo.edu (R. Nunkesser), schettlinger@statistik.uni-dortmund.de (K. Schettlinger).

0167-9473/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2006.12.039

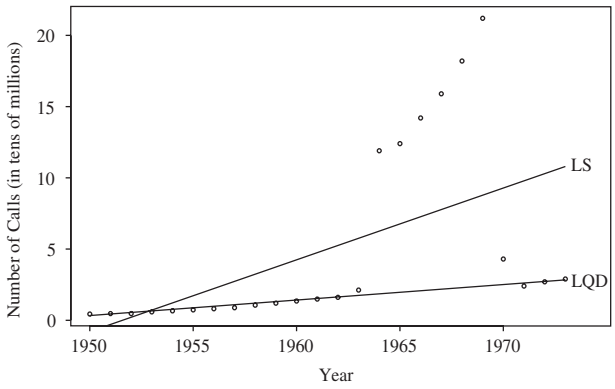

Fig. 1. An example for an LS and an LQD fit for data consisting of the number of international phone calls originated in Belgium between 1950 and 1973 (see Rousseeuw and Leroy, 1987). Partly in 1963 and 1970 and from 1964 to 1969 the duration of the calls was recorded instead of the number of calls.

fit. Also, 50% represents an upper bound for the breakdown point in the class of regression-equivariant estimators. An example for the importance of a high breakdown point is given in Fig. 1.

Further, the LQD estimator shows a much better performance at Gaussian distributed errors than other maximum breakdown methods such as *least median of squares* (LMS) (Rousseeuw, 1984) or *least trimmed squares* (LTS) regression (Rousseeuw and Leroy, 1987). *Deepest regression* (DR) (Rousseeuw and Hubert, 1999) shows similar behavior as the LQD for normally distributed samples but does not have a maximum breakdown value. As a drawback, robust regression methods generally need more computation time than non-robust methods. To make the above-mentioned methods feasible for practical applications, some research has been carried out for enhancing their computational speed and for geometrical interpretations of the regression problem (see e.g. Edelsbrunner and Souvaine, 1990; Mount et al., 1997 for LMS, Rousseeuw and Van Driessen, 2002 for LTS, and Langerman and Steiger, 2003 for DR).

For the definition of the LQD estimator, consider a line $L : y = \beta x + \alpha$ with slope β and intercept α, and let

$$r_i(L) = y_i - \beta x_i - \alpha$$

denote the residual of the point $p_i = (x_i, y_i)$ with respect to the line L. Further, denote the residual difference of the points p_i and p_j by

$$r_{i,j}(L) = r_i(L) - r_j(L) = (y_i - y_j) - \beta(x_i - x_j).$$

For bivariate data, the LQD estimator, introduced by Croux et al. (1994), is defined as follows:

Definition 1. Consider n points $p_i = (x_i, y_i) \in \mathbb{R}^2$, and let $h = \lfloor (n+3)/2 \rfloor$. The *LQD solution* to the regression problem is given by the slope of the line L which minimizes the $\binom{h}{2}$th order statistic of $\{|r_{i,j}(L)||1 \leqslant i < j \leqslant n\}$.

The intercept of the LQD regression fit with slope $\hat{\beta}$ has to be estimated afterwards, e.g. by $med\{y_i - \hat{\beta}x_i | 1 \leqslant i \leqslant n\}$. The exact algorithm Croux et al. propose needs time $\mathcal{O}(n^5 \log n)$. Another possibility to compute the LQD regression fit is to adapt LMS or *least quantile of squares* (LQS) algorithms. The adaption proposed by Croux et al. leads to a running time of $\mathcal{O}(n^4)$, if the algorithm of Edelsbrunner and Souvaine (1990) for LMS is used. Agulló (2002) proposes an approximation algorithm for LQD, but only gives empirical running time results.

Due to the high computational effort needed when using common algorithms, the LQD is not widely used yet. However, Dryden and Walker (1999) propose to use it for object matching in biology and Mebane and Sekhon (2004) use the LQD fit to detect outliers in vote counts.

A presentation of the LQD problem from the geometric point of view is stated in Section 2 while Sections 3 and Section 4 give a more detailed description of the single steps of the algorithms. In Section 5 the results are extended to LQS and LMS regression through the origin. Finally, Section 6 compares the running times of the described algorithms.

2. Solving the LQD geometrically

In their article, introducing the LQD estimator, Croux et al. (1994) propose to use the subset algorithm developed by Rousseeuw and Leroy (1987). This algorithm is based on examining subsets of the data points that determine local solutions. The $\binom{h}{2}$th order statistic of the absolute residual differences of a local solution can be computed in time $\mathcal{O}(n \log n)$. Croux et al. propose to examine all $\mathcal{O}(n^2)$ or alternatively just $\mathcal{O}(n)$ randomly chosen 2-subsets of the data points which needs overall time $\mathcal{O}(n^3 \log n)$ or $\mathcal{O}(n^2 \log n)$, respectively. However, the resulting algorithm is not exact because the global solution is not necessarily determined by a 2-subset. The exact algorithm they propose needs time $\mathcal{O}(n^5 \log n)$.

In contrast, we use the concept of geometric duality which Chazelle et al. (1985) propose for solving geometrical problems. In geometric duality points are mapped to lines and vice versa. Intuitively, it is easier to search for a point in an arrangement of lines, than to search for a line through a set of points. Moreover, we use that the LQD estimator is independent of the intercept and thus obtain an expected running time of $\mathcal{O}(n^2 \log n)$ and a deterministic running time of $\mathcal{O}(n^2 \log^2 n)$. We additionally state two easy to implement alternatives, an exact algorithm with expected running time $\mathcal{O}(n^2 \log^2 n)$ and an approximation algorithm with a running time of roughly $\mathcal{O}(n^2 \log n)$.

For convenience, we consider the axes of a space as ordered and introduce the terms 'below' and 'left' for the relation between a point and a line ('above', 'right', and 'on/intersecting' are defined analogously).

Definition 2. A line $y = \beta x + \alpha$ lies 'below' a point (x_p, y_p) in relation to the y-axis iff $\beta x_p + \alpha < y_p$. It lies 'left' of (x_p, y_p) in relation to the x-axis iff $(y_p - \alpha)/\beta < x_p$.

In order to solve the LQD problem geometrically we redefine it:

Definition 3 ($\mathrm{LQD}^{\mathrm{geom}}$ *problem*). Consider an input consisting of n points $(x_1, y_1), \ldots, (x_n, y_n)$, where $(x_i, y_i) \in \mathbb{R}^2$ and a positive integer h. Transforming the points for $1 \leqslant i < j \leqslant n$ to $2\binom{n}{2}$ lines

$$L^+_{i,j} : v = +(x_i - x_j)u - (y_i - y_j),$$

$$L^-_{i,j} : v = -(x_i - x_j)u + (y_i - y_j)$$

leads to a new space with axes u and v, which we call *modified dual space*. Now, the $\mathrm{LQD}^{\mathrm{geom}}$ problem consists of finding a point (β, r) such that $r \geqslant 0$ is minimal and $\binom{n}{2} + \binom{h}{2}$ lines are below (in relation to the v-axis) or intersecting it.

Definition 4. Each point on a line with k lines below or intersecting it is called a point on the *k-level*. If $k = \binom{h}{2} + \binom{n}{2}$, such a point is also called a *local solution*. Thus, the *global solution* of LQD^{geom} is the local solution with the minimum v-value in modified dual space.

We will show in the next lemma that an optimal LQD solution is obtained by solving the $\mathrm{LQD}^{\mathrm{geom}}$ problem. An example is given in Fig. 2.

Lemma 5. *Let $h = \lfloor (n+3)/2 \rfloor$. If the point (β, r) in modified dual space is an optimal solution of* $\mathrm{LQD}^{\mathrm{geom}}$, *then the LQD regression fit in primal space has slope β and the minimal $\binom{h}{2}$th order statistic of $\{|r_{i,j}(\beta x + \alpha)| | 1 \leqslant i < j \leqslant n\}$ is r for arbitrary intercept α.*

Proof. Let (β, r) be an optimal solution of $\mathrm{LQD}^{\mathrm{geom}}$ and consider arbitrary i and j with $1 \leqslant i < j \leqslant n$ and the corresponding lines $L^+_{i,j}$ and $L^-_{i,j}$. Now, consider the following three cases:

(1) $L^+_{i,j}$ and $L^-_{i,j}$ are below or intersecting (β, r).
(2) One line of $L^+_{i,j}$ and $L^-_{i,j}$ is above (β, r) and the other line is below or intersecting (β, r).
(3) $L^+_{i,j}$ and $L^-_{i,j}$ are above (β, r).

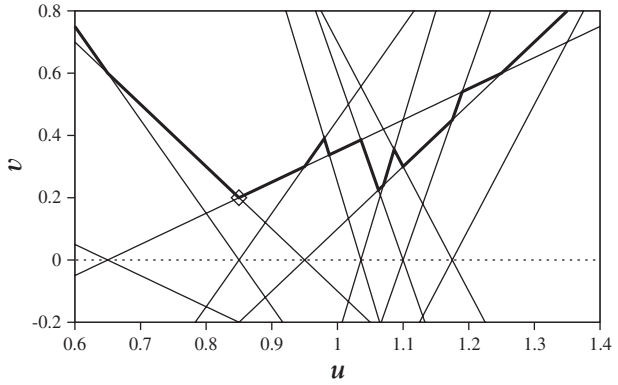

Fig. 2. An example for the mapping of the points {(0, 0.15), (1, 0.8), (3, 2.7), (7, 7.4)} to the corresponding 12 lines in the modified dual space. The LQD solution for $h = 3$ is determined by the lowest point with nine lines below or intersecting, here: (0.85, 0.2) (marked with $\diamond$). The bold lines show local solutions. The LQD regression line for a zero-intercept model in primal space is therefore $y = 0.85x$, and the corresponding third order statistic of the absolute residual differences takes on its minimal value of 0.2.

The third case does not occur, since $r \geqslant 0$ and $L^+_{i,j}$ and $L^-_{i,j}$ intersect on the u-axis. In the first case, the stated relations translate to the original problem as follows:

$$\begin{aligned} & L^+_{i,j} \text{ and } L^-_{i,j} \text{are below or intersecting } (\beta, r) \\ & \overset{\text{Def.2}}{\Leftrightarrow} \ (x_i - x_j)\beta - (y_i - y_j) \leqslant r \text{ and } -(x_i - x_j)\beta + (y_i - y_j) \leqslant r \\ & \Leftrightarrow \ |(x_i - x_j)\beta - (y_i - y_j)| \leqslant r \\ & \Leftrightarrow \ \text{For all intercepts } \alpha : |r_{i,j}(\beta x + \alpha)| \leqslant r. \end{aligned} \tag{1}$$

Now, recall that there are $\binom{n}{2} + \binom{h}{2}$ lines below or intersecting (β, r). Because of counting arguments, there are at least $\binom{h}{2}$ pairs (i, j) such that both lines $L^+_{i,j}$ and $L^-_{i,j}$ are below or intersecting (β, r). Due to Eq. (1), we obtain at least $\binom{h}{2}$ absolute residual differences smaller than or equal to r with respect to an arbitrary line with slope β. In addition, $r \geqslant 0$ is the minimal value such that (β, r) has $\binom{n}{2} + \binom{h}{2}$ lines below or intersecting it. Therefore, at most $\binom{n}{2} + \binom{h}{2} - 1$ lines are strictly below (β, r) (the point (β, r) has to be located on a line) and due to counting arguments at most $\binom{h}{2} - 1$ absolute residual differences are strictly smaller than r. Hence, r is the $\binom{h}{2}$th order statistic of $\{|r_{i,j}(\beta x + \alpha)| | 1 \leqslant i < j \leqslant n\}$.

We claim that no other line $y = \beta' x + \alpha$ leads to a smaller $\binom{h}{2}$th order statistic r'. Assume for the sake of contradiction that there is a slope β' leading to a smaller $\binom{h}{2}$th order statistic r'. Due to Eq. (1), there are $2\binom{h}{2}$ lines $L^+_{i,j}$ and $L^-_{i,j}$ below or intersecting (β', r'). Of the remaining $2\binom{n}{2} - 2\binom{h}{2}$ lines at least $\left(2\binom{n}{2} - 2\binom{h}{2}\right)/2$ lines are also below or intersecting (β', r') (recall that case three does not occur). Thus, (β', r') is a solution to LQD^{geom} with $r' < r$, which is a contradiction, because (β, r) is the global solution to LQD^{geom} (and therefore the local solution with the smallest v-coordinate). Hence, $L : y = \beta x + \alpha$ minimizes the $\binom{h}{2}$th order statistic of $\{|r_{i,j}(L)| | 1 \leqslant i < j \leqslant n\}$. □

Note that the transformation of the input in LQD^{geom} needs time $\mathcal{O}(n^2)$.

Theorem 6. *It is possible to compute the LQD estimator for n data points in the plane in expected running time* $\mathcal{O}(n^2 \log n)$ *or deterministic running time* $\mathcal{O}(n^2 \log^2 n)$.

Proof. For arbitrary lines, the problem of finding the lowest point on the k-level is also known as *minimum k-level point* or *k-violation linear programming*. For m lines this problem is solved with *parametric search* (see Megiddo, 1979) by

Cole et al. (1987) and by Roos and Widmayer (1994) in time $\mathcal{O}(m \log^2 m)$. Chan (1999) states a randomized algorithm using *cuttings* that needs expected time $\mathcal{O}(m \log m)$. As we consider $\mathcal{O}(n^2)$ lines, the theorem is proved. □

The shortcomings of the algorithms proposed in Chan (1999), Cole et al. (1987), and Roos and Widmayer (1994) are the rather complicated techniques involved (see e.g. Agarwal and Sharir, 1998; van Oostrum and Veltkamp, 2002 for drawbacks of *parametric search* and e.g. Section 4 of Har-Peled, 1998 for problems in the implementation of *cutting* algorithms). To overcome these shortcomings, we provide two easy to implement algorithms with similar running times.

Now, recall the transformation of the original data. The transformed data lie in the modified dual space where the LQD solution of the original regression problem is represented by a point. All local solutions with the same value r are located on a horizontal line. Using this fact, the transformation to the LQD$^{\text{geom}}$ problem enables us to use a method that decides in time $\mathcal{O}(n^2 \log n)$ whether a given value r belongs to a local solution. This decision method is presented in the next section. Hereinafter, we refer to this method as DECIDELQD and DECIDELQD(r) if r is the given value.

In a second step, we propose two algorithms that solve the LQD$^{\text{geom}}$ problem using DECIDELQD in Section 4.

3. Solving the decision problem

In the following, we specify the method DECIDELQD—used in the next sections for solving the underlying decision problem—in PDL (see e.g. Caine and Gordon, 1975). Given a set of lines S (in our case consisting of all $2\binom{n}{2}$ lines in modified dual space) and a height r (a fixed v-coordinate in the modified dual space), we need to decide whether there exists a point at this height with $\binom{h}{2} + \binom{n}{2}$ lines below or intersecting it. Let $\mathcal{H}(r)$ denote the horizontal line at height r.

Algorithm 1.

DECIDELQD (*set of lines S*, *height r*).
Compute all intersections of the line $\mathcal{H}(r)$ *with the lines in S*
Sort the intersections from left to right, *intersections with the same value*
  are ordered ascending according to the slope of the intersecting line
Determine the number of lines below or on the first point in the sorted set
  and subtract 1 *if the slope of the line intersecting the first point is negative*
Initialize the count of lines below or on the current point with this number
do for *each intersection point in sorted order*
  if *the slope of the intersecting line is negative*
    Increase the count of lines lying below or on the current point by one
  else
    Decrease the count of lines lying below or on the current point by one
  endif
  if *the count of these lines is at least* $\binom{h}{2} + \binom{n}{2}$
    return true, *i.e. a local solution exists at height r*
  endif
enddo
return false, *i.e. no local solution exists at height r or a smaller height*.

Lemma 7. *The algorithm* DECIDELQD *decides in time* $\mathcal{O}(n^2 \log n)$ *whether there exists a point at a given height in the modified dual space with* $\binom{h}{2} + \binom{n}{2}$ *lines below or intersecting it.*

Proof. Of all steps, sorting costs most time, namely $\mathcal{O}(n^2 \log n)$. □

Note, that DECIDELQD can additionally report the encountered local solution if necessary. We will need this in the following algorithms to return the solution found.

4. Searching for the optimal point

To search for the optimal point in the modified dual space we propose two methods, which lead to two different algorithms:

(1) A deterministic search based on the geometric mean to get an approximative solution.
(2) A randomized search leading to a Las Vegas algorithm.

In both proposed methods we denote the upper bound for the height of the optimal solution by $r_{\max}$ and the lower bound by $r_{\min}$. To obtain an approximative solution with *approximation ratio* $1+\varepsilon$ the inequation $r_{\max}/r_{\min} \leqslant 1+\varepsilon$ has to hold.

Algorithm 2.

APPROXIMATIVESEARCH (*set of lines S, approximation ratio* ε).
Initialize $r_{\max}$ *as* ∞ *and* $r_{\min}$ *as* 0
Test the heights 0, $1/(1+\varepsilon)$, 1, *and* $1+\varepsilon$ *for local solutions with* DECIDELQD
Update $r_{\min}$ *to the greatest value that* DECIDELQD *returned* ***false*** *for*
Update $r_{\max}$ *to the smallest value that* DECIDELQD *returned* ***true*** *for*
if $r_{\max}$ *is still* ∞
  do while DECIDELQD(S,$r_{\min}^2$) *returns* ***false***
    $r_{\min} := r_{\min}^2$
  enddo
  $r_{\max} := r_{\min}^2$
elseif $r_{\min}$ *is still* 0
  do while DECIDELQD(S,$\sqrt{r_{\max}}$) *returns* ***true***
    $r_{\max} := \sqrt{r_{\max}}$
  enddo
  $r_{\min} := \sqrt{r_{\max}}$
endif
do while *the desired approximation ratio* $1+\varepsilon$ *is not reached*
  if DECIDELQD(S,$\sqrt{r_{\min} r_{\max}}$) *returns* ***true***
    $r_{\max} := \sqrt{r_{\min} r_{\max}}$
  else
    $r_{\min} := \sqrt{r_{\min} r_{\max}}$
  endif
enddo
return *the local solution at height* $r_{\max}$.

Theorem 8. *The approximation algorithm finds the LQD fit with approximation ratio* $1+\varepsilon$ $(0<\varepsilon\leqslant 1)$ *on n points in the plane in worst case time*:

$$\begin{cases} \mathcal{O}\left(n^2 \log n \, \log \log_{(1+\varepsilon)} \max\left\{\dfrac{1}{r^*}, r^*\right\}\right) & \text{whenever } \max\left\{\dfrac{1}{r^*}, r^*\right\} > 1+\varepsilon, \\ \mathcal{O}(n^2 \log n) & \text{otherwise}, \end{cases}$$

where r^* *is the* $\binom{h}{2}$*th order statistic of the absolute residual differences of the LQD fit.*

Proof. The correctness of the algorithm follows from the fact that solving $\mathrm{LQD}^{\mathrm{geom}}$ suffices to obtain an LQD fit due to Lemma 5 and that DECIDELQD is correct due to Lemma 7. As the running time is bounded (we show this below), the algorithm terminates.

Observe, that r^* is also the height of the optimal solution. If $r^*=0$ or $r^* \in [1/(1+\varepsilon), 1+\varepsilon] \Leftrightarrow \max\{1/r^*, r^*\} \leqslant 1+\varepsilon$, the four initial tests at heights 0, $1/(1+\varepsilon)$, 1, and $1+\varepsilon$ suffice to attain the desired approximation ratio.

If $\max\{1/r^*, r^*\} > 1+\varepsilon$ either $r_{\max}$ is still ∞ or $r_{\min}$ is still 0 after these tests. We only consider the case that $r_{\max}$ is still ∞, because the calculations for the other case are similar. If $r_{\max}$ is still ∞, $r_{\min}$ has to be the greatest of the initially

tested values, namely $1+\varepsilon$. After the ensuing do loop terminates $r_{\max} = r_{\min}^2$ (recall that $r_{\min}$ is updated if DECIDELQD returns false) and therefore $(r^*)^2 > r_{\max}$ (recall that $r_{\min} \leqslant r^* \leqslant r_{\max}$). Hence, the maximum number DECIDELQD is called to obtain $r_{\max}$ is determined by the smallest integer k_1 that is a solution to $(1+\varepsilon)^{2^{k_1}} \geqslant (r^*)^2$. Therefore, the maximum number DECIDELQD is called is $\lceil 2 \log\log r^* - \log\log(1+\varepsilon)\rceil$.

Since $r_{\max} = r_{\min}^2$, we obtain $r_{\max}/r_{\min} = r_{\min}$. In the last do loop the geometric mean of $r_{\min}$ and $r_{\max}$ is tested until the approximation ratio $1+\varepsilon$ is reached. In each loop cycle, we obtain new bounds $r_{\min}$ and $r_{\max}$. One is identical to the former bound, the other is the geometric mean of the former bounds. For the case that $r_{\max}$ is updated, the new ratio between $r_{\max}$ and $r_{\min}$ is

$$\frac{r_{\max}}{r_{\min}} = \frac{\sqrt{r'_{\max} r_{\min}}}{r_{\min}} = \sqrt{\frac{r'_{\max}}{r'_{\min}}},$$

where $r'_{\min}$ and $r'_{\max}$ denote the old values of $r_{\min}$ and $r_{\max}$, respectively. The other case leads to the same ratio. Hence, the new ratio is the square root of the old ratio. Since the ratio we begin with is less than r^*, the maximum number DECIDELQD is called until we reach a ratio of $1+\varepsilon$ is determined by the smallest integer k_2 that is a solution to $(r^*)^{(1/2)^{k_2}} \leqslant 1+\varepsilon$. Therefore, the maximum number DECIDELQD is called is $\lceil \log\log r^* - \log\log(1+\varepsilon)\rceil$. Each call of DECIDELQD costs time $\mathcal{O}(n^2 \log n)$. □

For the randomized algorithm, we need the following definition.

Definition 9. An *inversion* in a permutation π is a pair of values where $i > j$ and $\pi(i) < \pi(j)$ or $i < j$ and $\pi(i) > \pi(j)$. An *inversion table* contains the number of inversions for each element i, denoted by inv(i).

We denote the horizontal line at height r by $\mathcal{H}(r)$.

Algorithm 3.

RANDOMIZEDSEARCH (*set of lines S*).
Initialize $r_{\min}$ *as* 0
Randomly choose a u-coordinate where a line intersects the u-axis
Initialize $r_{\max}$ *as the height of the best solution with this u-coordinate*
do until *there are no more intersections between* $\mathcal{H}(r_{\min})$ *and* $\mathcal{H}(r_{\max})$
  Calculate the number of intersections $\mathcal{I}$ *between* $\mathcal{H}(r_{\min})$ *and* $\mathcal{H}(r_{\max})$ *by*
    Computing all intersections between the lines in S and $\mathcal{H}(r_{\min})$
    Computing all intersections between the lines in S and $\mathcal{H}(r_{\max})$
    Labelling the lines in S according to their intersections on $\mathcal{H}(r_{\min})$
      from left to right
    Interpreting the intersections on $\mathcal{H}(r_{\max})$ *as a permutation of the labels*
    Computing the inversion table of this permutation
    Summing up the entries in the inversion table (*this yields* $\mathcal{I}$)
  Randomly choose one of the intersections between $\mathcal{H}(r_{\min})$ *and* $\mathcal{H}(r_{\max})$ *by*
    Choosing a line ℓ *in S, where line i is chosen with probability* $inv(i)/\mathcal{I}$
    Choosing an intersection on ℓ *between* $\mathcal{H}(r_{\min})$ *and* $\mathcal{H}(r_{\max})$
      uniformly at random
  Let r_{mid} *be the height of this intersection*
  if DECIDELQD(*S*,r_{mid}) *returns* ***true***
    $r_{\max} := r_{\text{mid}}$
  else
    $r_{\min} := r_{\text{mid}}$
  endif
enddo
return *the local solution at height* $r_{\max}$.

Theorem 10. *The randomized algorithm finds the LQD fit on n points in the plane in expected running time of* $\mathcal{O}(n^2 \log^2 n)$.

Proof. The algorithm is correct due to Lemmas 5 and 7, and because each entry in the inversion table contains the number of intersections on the corresponding line that lies between $\mathcal{H}(r_{\min})$ and $\mathcal{H}(r_{\max})$. In the following, we show that the expected running time of the algorithm is $\mathcal{O}(n^2 \log^2 n)$.

It is possible to compute a starting value for $r_{\max}$ by computing the best local solution for a fixed u-coordinate in time $\mathcal{O}(n^2 \log n)$. To do this, we have to calculate all intersections of the lines with the chosen u-coordinate, sort them according to their v-coordinate, and sift through the intersection points increasing the count of lines below until we reach an intersection with $\binom{h}{2} + \binom{n}{2}$ lines below or on it.

The calculation of the number of intersections between $v = r_{\min}$ and $r_{\max}$ is possible in time $\mathcal{O}(n^2 \log n)$, because the inversion table can be computed in this time (for example, with an extended merge sort algorithm) and no other step takes more time than $\mathcal{O}(n^2 \log n)$. Randomly choosing an intersection between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$ can also be done in time $\mathcal{O}(n^2 \log n)$, because computing the intersections on the randomly choosen line is the costliest step.

Let $\mathcal{I}(\mathcal{H}(r_1), \mathcal{H}(r_2))$ denote the number of intersections between two horizontal lines $\mathcal{H}(r_1)$ and $\mathcal{H}(r_2)$ and let $m := \mathcal{I}(\mathcal{H}(r_{\min}), \mathcal{H}(r_{\max}))$. Furthermore, assume that no two intersections have the same v-coordinate. The algorithm chooses each intersection between $\mathcal{H}(r_{\min})$ and $\mathcal{H}(r_{\max})$ with probability $1/m$, therefore the expected number of remaining intersections is

$$E(\mathcal{I}(\mathcal{H}(r_{\min}), \mathcal{H}(r_{\mathrm{mid}}))) = E(\mathcal{I}(\mathcal{H}(r_{\mathrm{mid}}), \mathcal{H}(r_{\max}))) = \sum_{i=0}^{m-1} \frac{1}{m} i < \frac{m}{2}.$$

Intersections with the same v-coordinate can only lead to a smaller value. Hence, we expect to execute $\mathcal{O}(\log n)$ cycles of the do loop and expect an overall running time of $\mathcal{O}(n^2 \log^2 n)$. □

5. LQS and LMS regression through the origin

To compute an LQS regression through the origin, the same geometric interpretation as in Section 2 can be used—with $\mathcal{O}(n)$ instead of $\mathcal{O}(n^2)$ lines. The LQS estimator was introduced by Rousseeuw and Leroy (1987) and later redefined by Rousseeuw and Hubert (1997) as follows:

Definition 11. Consider n points $p_i = (x_i, y_i) \in \mathbb{R}^2$ and a positive integer h with $2 \leqslant h \leqslant n$. The *LQS solution* to the regression problem is given by the line L which minimizes the hth order statistic of the absolute residuals $\{|r_i(L)||1 \leqslant i \leqslant n\}$.

The LMS estimator introduced by Rousseeuw (1984) is a special case of the LQS estimator, as it minimizes the median of the absolute residuals. It is easy to see that a transformation of the n points $p_i = (x_i, y_i)$ to the $2n$ lines $L_i^+ : v = +x_i u - y_i$ and $L_i^- : v = -x_i u + y_i$ enables us to compute LQS and LMS regression through the origin with the algorithms proposed here.

Corollary 12. *It is possible to compute LQS and LMS regression through the origin for n bivariate data points in expected running time* $\mathcal{O}(n \log n)$ *or deterministic running time* $\mathcal{O}(n \log^2 n)$.

This improves a result of Barreto and Maharry (2006), who stated an algorithm with running time $\mathcal{O}(n^2 \log n)$ for LMS regression through the origin.

6. Experimental results

In contrast to Chan (1999), Cole et al. (1987), and Roos and Widmayer (1994), where no implementation of the algorithms in Theorem 6 is mentioned, we have implemented the algorithms presented in Section 4. While it is theoretically possible to choose ε in such a way that the approximation algorithm is slower than the randomized algorithm, trial runs with our implementations show that the approximative version is generally faster in practice. For the conducted experiments, we used 64 bit floating point numbers according to IEEE 754-1985. Note, that

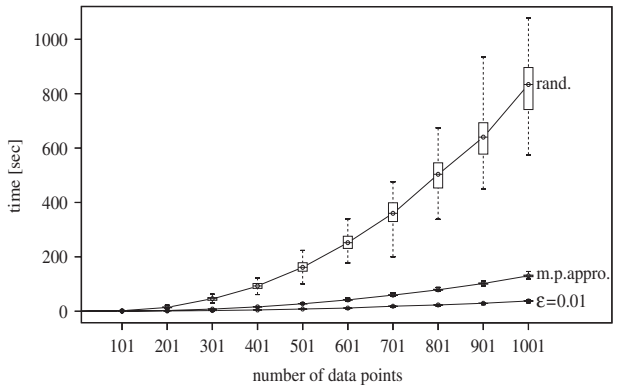

Fig. 3. Boxplots of the running time in seconds on a Pentium 4 CPU with 2.56 GHz and 1024 MB of RAM for the first type of data set.

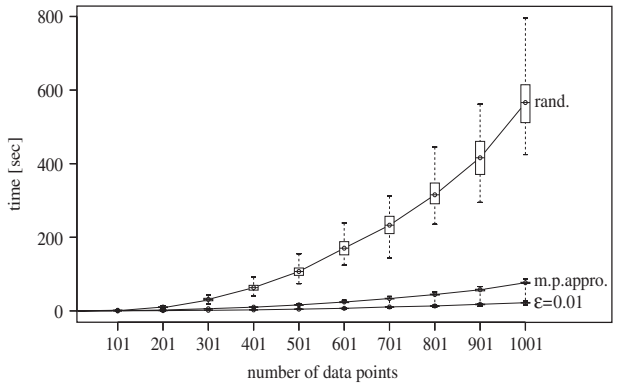

Fig. 4. Boxplots of the running time in seconds on a Pentium 4 CPU with 2.56 GHz and 1024 MB of RAM for the second type of data set.

Har-Peled (1998) had to use rational numbers when trying to implement algorithms for *cuttings*, which are needed for the algorithm of Chan (1999). If we choose ε sufficiently small and wait until $r_{\min}$ and $r_{\max}$ are indistinguishable from their geometric mean the approximative version computes the same results as the randomized version (except for possible rounding errors).

The experiments show that even with such a precision, the approximative version is faster than the randomized one. However, for greater ε it is of course much faster. We compare the approximative version with maximal precision for 64 bit floating point numbers to the approximative version with $\varepsilon = 0.01$ and to the randomized version on two types of data sets with n points. The first type of data set is

$$\left\{(x_i, y_i)|x_i = \frac{2(i-1)}{n-1};\ y_i = -x_i + 1.2 + e_1;\ 1 \leqslant i \leqslant n\right\}$$

and the second is

$$\begin{cases} \left\{(x_i, y_i)|x_i = \dfrac{2(i-1)}{n-1};\ y_i = e_2;\ 1 \leqslant i \leqslant n\right\} & \text{whenever } i \leqslant \left\lfloor \dfrac{n}{2} \right\rfloor + 1, \\ \left\{(x_i, y_i)|x_i = \dfrac{2(i-1)}{n-1};\ y_i = -\dfrac{1}{10}x_i + \dfrac{3}{2} + e_2;\ 1 \leqslant i \leqslant n\right\} & \text{otherwise,} \end{cases},$$

where e_1 is random noise from a normal distribution with mean 0 and standard deviation 10^{-2}, and e_2 is random noise from a normal distribution with mean 0 and standard deviation 10^{-280}. While the first type of data set represents uncontaminated normal data, the second type contains $\lceil n/2 \rceil - 1$ outliers. Thus, data set number two can result in local solutions that are far from the optimum.

Computing times of these three versions of the algorithm are measured for each n in $\{101, 201, \ldots, 1001\}$ for 100 different data sets. The results for the first type of data set are shown in Fig. 3, the outcomes for the second type are

shown in Fig. 4. The figures show boxplots of the running times for each n and each algorithm. These boxplots illustrate the minimal and maximal running time for each n as well as the quartiles and the median of the running times. The medians are connected by additional lines.

It clearly shows that the randomized version has a considerably larger variance in its computational time, and needs much more time than the approximative version. Another noticeable fact is that the two figures do not differ very much. The high number of outliers and local solutions in the second data set does not slow down the algorithms. On the contrary, the possibility to start at a local solution that is far below other local solutions leads to better performance. This is also the reason for the long lower whiskers of the boxplots for the approximation algorithm with maximum precision.

In conclusion, the randomized version of the algorithms presented in Section 4 provides a large improvement in computational time on currently available LQD algorithms. However, the experiments show that the proposed approximation algorithm yields even better results. Therefore, these algorithms might increase the practical relevance of LQD regression in the future.

References

Agarwal, P.K., Sharir, M., 1998. Efficient algorithms for geometric optimization. ACM Comput. Surv. 30 (4), 412–458.

Agulló, J., 2002. An exchange algorithm for computing the least quartile difference estimator. Metrika 55, 3–16.

Barreto, H., Maharry, D., 2006. Least median of squares and regression through the origin. Comput. Statist. Data Anal. 50, 1391–1397.

Caine, S.H., Gordon, E.K., 1975. Pdl—a tool for software design. In: Proceedings of the National Computer Conference, pp. 271–276.

Chan, T.M., 1999. Geometric applications of a randomized optimization technique. Discrete Comput. Geom. 22 (4), 547–567.

Chazelle, B., Guibas, L.J., Lee, D.T., 1985. The power of geometric duality. BIT 25 (1), 76–90.

Cole, R., Sharir, M., Yap, C.K., 1987. On k-hulls and related problems. SIAM J. Comput. 16 (1), 61–77.

Croux, C., Rousseeuw, P.J., Hössjer, O., 1994. Generalized S-estimators. J. Amer. Statist. Assoc. 89, 1271–1281.

Donoho, D., Huber, P., 1983. The notion of breakdown point. In: Bickel, P., Doksum, K., Hodges, J.J. (Eds.), A Festschrift for Erich L. Lehmann. Wadsworth, Belmont, CA, pp. 157–184.

Dryden, I.L., Walker, G., 1999. Highly resistant regression and object matching. Biometrics 55 (3), 820–825.

Edelsbrunner, H., Souvaine, D., 1990. Computing least median of squares regression and guided topological sweep. J. Amer. Statist. Assoc. 85, 115–119.

Har-Peled, S., 1998. Constructing cuttings in theory and practice. In: SCG '98: Proceedings of the 14th Annual Symposium on Computational Geometry.ACM Press, New York, pp. 327–336.

Langerman, S., Steiger, W.L., 2003. The complexity of hyperplane depth in the plane. Discrete Comput. Geom. 30 (2), 299–309.

Mebane Jr., W.R., Sekhon, J.S., 2004. Robust etimation and outlier detection for overdispersed multinomial models of count data. Amer. J. of Political Sci. 48 (2), 391–410.

Megiddo, N., 1979. Combinatorial optimization with rational objective functions. Math. Oper. Res. 4 (4), 414–424.

Mount, D.M., Netanyahu, N.S., Romanik, K., Silverman, R., Wu, A.Y., 1997. A practical approximation algorithm for the LMS line estimator. In: SODA '97. SIAM, Philadelphia, PA, pp. 473–482.

Roos, T., Widmayer, P., 1994. k-violation linear programming. Inform. Process. Lett. 52 (2), 109–114.

Rousseeuw, P.J., Van Driessen, K., 2002. Computing LTS regression for large data sets. Estadística 54, 163–190.

Rousseeuw, P.J., 1984. Least median of squares regression. J. Amer. Statist. Assoc. 79, 871–880.

Rousseeuw, P.J., Hubert, M., 1997. Recent developments in PROGRESS. in: Dodge, Y. (Ed.), L_1-Statistical Procedures and Related Topics. Lecture Notes-Monograph Series, vol. 31. Institute of Mathematical Statistics, pp. 201–214.

Rousseeuw, P.J., Hubert, M., 1999. Regression depth. J. Amer. Statist. Assoc. 94 (446), 388–402.

Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. Wiley, New York.

van Oostrum, R., Veltkamp, R.C., 2002. Parametric search made practical. In: SCG '02: Proceedings of the 18th Annual Symposium on Computational Geometry.ACM Press, New York, NY, USA, pp. 1–9.