Bernholt, Thorsten; Nunkesser, Robin; Schettlinger, Karen

Working Paper

Computing the Least Quartile Difference Estimator in the Plane

Technical Report, No. 2005,51

Provided in Cooperation with:
Collaborative Research Center 'Reduction of Complexity in Multivariate Data Structures' (SFB 475), University of Dortmund

Suggested Citation: Bernholt, Thorsten; Nunkesser, Robin; Schettlinger, Karen (2005) : Computing the Least Quartile Difference Estimator in the Plane, Technical Report, No. 2005,51, Universität Dortmund, Sonderforschungsbereich 475 - Komplexitätsreduktion in Multivariaten Datenstrukturen, Dortmund

This Version is available at:
https://hdl.handle.net/10419/22644

Computing the Least Quartile Difference Estimator in the Plane*

Thorsten Bernholt[1], Robin Nunkesser[1], and Karen Schettlinger[2]

[1] FB Informatik, LS2, Universität Dortmund, 44221 Dortmund, Germany
{thorsten.bernholt,robin.nunkesser}@udo.edu
[2] FB Statistik, Universität Dortmund, 44221 Dortmund, Germany
schettlinger@statistik.uni-dortmund.de

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, e.g. the subset algorithm of Rousseeuw and Leroy. For computing the LQD estimator for n data points in the plane, we propose a randomized algorithm with expected running time $\mathcal{O}(n^2 \log^2 n)$ and an approximation algorithm with a running time of roughly $\mathcal{O}(n^2 \log n)$. It can be expected that the practical relevance of the LQD estimator will strongly increase thereby.

1 Introduction

Finding relationships between different variables is a common and multidisciplinary problem. Assuming a linear dependence between two variables, this relationship can be estimated by applying linear regression methods.
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 [4] 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, Rousseeuw and Hössjer [3], has a breakdown point of $\lfloor n/2 \rfloor / n$ if the data fulfil certain requirements. This means, that up to 50% of the data can be contaminated without ruining the 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.

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

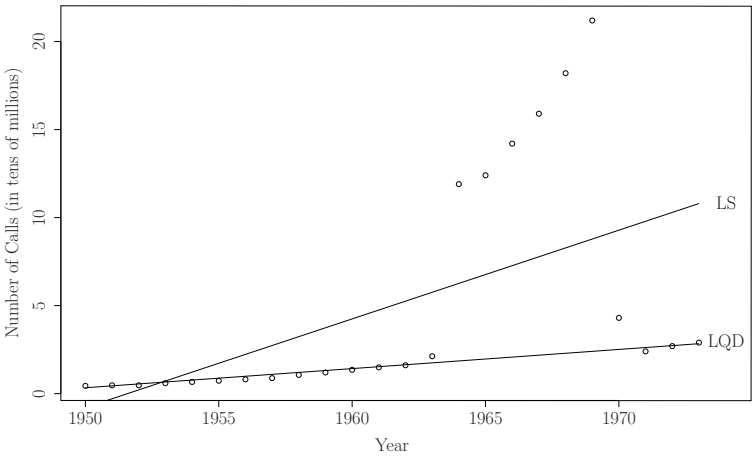

Fig. 1. An example for an LS and an LQD fit for data consisting of the number of international phone calls originated in Belgium between 1950 and 1973 (see Rousseeuw and Leroy [11]). Partly in 1963 and 1970 and from 1964 to 1969 the duration of the calls was recorded instead of the number of calls.

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) [9] or *least trimmed squares* (LTS) regression [11]. *Deepest regression* (DR) [10] shows similar behaviour 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. [6] and [8] for LMS, [12] for LTS, and [7] 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

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

For data in the plane, the *least quartile difference* (LQD) estimator, introduced by Croux, Rousseeuw and Hössjer [3], 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 $\{|r_{i,j}\,(L)| \mid 1 \le i < j \le n\}$.*

The intercept of the LQD regression fit with slope $\hat{\beta}$ has to be estimated afterwards, e.g. by $med\{y_i - \hat{\beta} x_i \mid 1 \le i \le n\}$. The exact algorithm Croux, Rousseeuw and Hössjer 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, Rousseeuw and Hössjer leads to a running time of $\mathcal{O}(n^4)$, if the algorithm of Edelsbrunner and Souvaine [6] for LMS is used. Agulló [1] 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 [5] propose to use it for object matching in biology and Mebane, Sekhon and Wand [13] 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 Sect. 2 while Sect. 3 and Sect. 4 give a more detailed description of the single steps of the algorithms. Finally, Sect. 5 compares the running times of the described algorithms.

2 Solving the LQD geometrically

In their article, introducing the LQD estimator, Croux, Rousseeuw and Hössjer [3] propose to use the subset algorithm developed by Rousseeuw and Leroy [11]. 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, Rousseeuw and Hössjer 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, Guibas and Lee [2] propose for solving geometrical problems. Hence, we obtain an expected running time of $\mathcal{O}(n^2 \log^2 n)$ for our exact algorithm and a running time of roughly $\mathcal{O}(n^2 \log n)$ for our approximation algorithm.
In order to solve the LQD problem geometrically, we redefine it:

Definition 2 (LQD$^{\mathbf{geom}}$ problem). *Consider an input consisting of n points $(x_1, y_1), \ldots, (x_n, y_n)$ in the plane and a positive integer h. Transforming the points for $1 \leq i < j \leq n$ 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 LQD^{geom} problem consists of finding a point (β, r), such that $r \geq 0$ is minimal and $\binom{n}{2} + \binom{h}{2}$ lines are below (in relation to the v-axis) or intersecting it.*

Definition 3. *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 LQD$^{\text{geom}}$ problem. An example is given in Fig. 2.

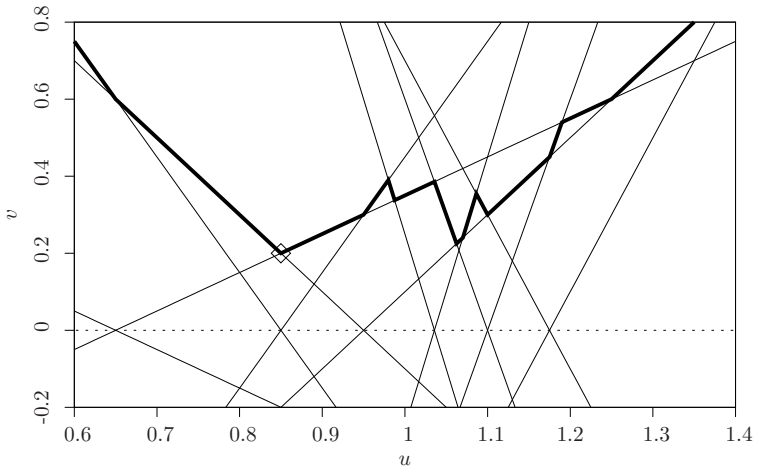

Fig. 2. An example for the mapping of the points $\{(0, 0.15), (1, 0.8), (3, 2.7), (7, 7.4)\}$ to the corresponding 12 lines in the modified dual space. The LQD solution for $h = 3$ is determined by the lowest point with nine lines below or intersecting, here: $(0.85, 0.2)$ (marked with $\diamond$).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

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

Proof. Let (β, r) be an optimal solution of LQD$^{\text{geom}}$ and consider arbitrary i and j with $1 \le i < j \le n$ 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).

The third case does not occur, since $r \ge 0$ and $L^+_{i,j}$ and $L^-_{i,j}$ intersect on the u-axis. Recall, that a line $v = au + b$ is below or intersecting a point (β, r), iff $a\beta + b \le r$. In the first case, the stated relations translate to the original problem as follows:

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

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 Equation 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, $r \geq 0$ 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) ((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 $\{|r_{i,j}(\beta x + \alpha)| \mid 1 \leq i < j \leq n\}$.
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 Equation 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 $(2\binom{n}{2} - 2\binom{h}{2})/2$ lines are also below or intersecting (β', r') (recall, that case three does not occur). Thus, (β', r') is a solution to LQD$^{\text{geom}}$ with $r' < r$, which is a contradiction, because (β, r) is the global solution to LQD$^{\text{geom}}$ (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 $\{|r_{i,j}(L)| \mid 1 \leq i < j \leq n\}$. □

Note, that the transformation of the input in LQD$^{\text{geom}}$ needs time $\mathcal{O}(n^2)$. 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 Sect. 4.

3 Solving the Decision Problem

In the following, we give a more detailed description of the method DECIDELQD, used in the next sections for solving the underlying decision problem. With the term 'height' we refer to the size of the v-coordinate in the modified dual space. Given a height r, 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. Computing all intersections of $\mathcal{H}(r)$ with the lines in the modified dual space and sorting them, costs $\mathcal{O}\left(n^2 \log n\right)$ time. For the first intersection point in the sorted set of intersection points, the number of lines lying below or on it can be determined quickly: unless there are other horizontal lines - a horizontal line in our construction starts in level $\binom{n}{2}$. If there are other horizontal lines, we have to add the number of horizontal lines below or on $\mathcal{H}(r)$ to this level. This is possible in time $\mathcal{O}\left(n^2\right)$. We now sift through the intersection points from left to right

increasing or decreasing the count of subjacent lines, depending on whether the intersecting line has negative or positive slope, in time $\mathcal{O}\left(n^2\right)$. In this way, it is possible to decide whether a point with $\binom{h}{2} + \binom{n}{2}$ subjacent or incident lines exists for the given height, and, if it exists, to determine that point.

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.

In both proposed methods we denote the upper bound for the height of the optimal solution by $r_{\max}$ and the lower bound by $r_{\min}$.
We have shown in Sect. 3, that it is possible to decide with DECIDELQD whether a local solution exists at a given height and to determine this solution, if it exists, in time $\mathcal{O}\left(n^2 \log n\right)$. If DECIDELQD decides for some height r in $[r_{\min}, r_{\max}]$ that there is a local solution, we update $r_{\max}$ to r. If DECIDELQD decides that there is no local solution at height r, we update $r_{\min}$ to r.
First, it is tested with DECIDELQD(0) whether a trivial solution with residual difference 0 exists to assure that $r_{\min} > 0$ when we first determine it. If a trivial solution exists, the algorithm stops and outputs the point determined in the test.
Otherwise, we continue to search for the optimal solution. We propose two algorithms for this, which are described in detail in the next two sections.

4.1 Approximative Search

For a solution with approximation ratio $1+\varepsilon$ the inequation $r_{\max}/r_{\min} \leq 1+\varepsilon$ has to hold. If such a solution is found, the approximation algorithm outputs the local solution determined by DECIDELQD($r_{\max}$). Our method to achieve $r_{\max}/r_{\min} \leq 1+\varepsilon$ is to iteratively calculate the geometric mean $\sqrt{r_{\max} r_{\min}}$, decide with DECIDELQD whether a local solution exists at this height in time $\mathcal{O}\left(n^2 \log n\right)$, and change $r_{\min}$ or $r_{\max}$ to $\sqrt{r_{\max} r_{\min}}$, depending on the decision of DECIDELQD until $r_{\max}/r_{\min} \leq 1+\varepsilon$. However, this is not possible as long as no values for $r_{\max}$ and $r_{\min}$ are known. To obtain an initial value for $r_{\min} > 0$, we test with DECIDELQD, whether a local solution at the heights $1/(1+\varepsilon)$, 1 and $1+\varepsilon$ exists. Therewith, we can decide, whether a solution in $[1/(1+\varepsilon), 1]$ or $[1, 1+\varepsilon]$ exists. In that case, we obtain a local solution at the height 1 or $1+\varepsilon$ and the desired approximation ratio $r_{\max}/r_{\min} \leq 1+\varepsilon$ is reached. Otherwise we either obtain an upper bound $r_{\max} = 1/(1+\varepsilon)$ or a lower bound $r_{\min} = 1+\varepsilon$. We only consider the case $r_{\min} = 1+\varepsilon$, because the calculations for the other case are similar (mostly we have to use the reciprocals of the values in this case). We determine $r_{\max}$ by iteratively squaring $r_{\min}$, testing for local solutions with DECIDELQD, and updating $r_{\min}$ until we find a height where a local solution exists.

Let r^* be the height of the optimal solution. After $r_{\max}$ is determined, $r_{\max} = r_{\min}^2$ and therefore $(r^*)^2 > r_{\max}$. The maximum number of steps to obtain $r_{\max}$ is determined by the smallest integer k_1 that is a solution to $(1+\varepsilon)^{2^{k_1}} \geq (r^*)^2$. Therefore, the maximum number of steps is $\lceil 2 \log\log r^* - \log\log(1+\varepsilon) \rceil$. Since $r_{\max} = r_{\min}^2$, we obtain $r_{\max}/r_{\min} = r_{\min}$. We now use the geometric mean as described above to determine better values for $r_{\min}$ and $r_{\max}$, respectively. In each step, 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 of steps to reach a ratio of $1+\varepsilon$ is determined by the smallest integer k_2 that is a solution to $(r^*)^{(1/2)^{k_2}} \leq 1+\varepsilon$. Therefore, the maximum number of steps is $\lceil \log\log r^* - \log\log(1+\varepsilon) \rceil$. All in all, the approximative search needs $\mathcal{O}(\log\log r^* - \log\log(1+\varepsilon))$ steps in the considered case. As each of these steps takes time $\mathcal{O}\left(n^2 \log n\right)$, we obtain the following:

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

$$\begin{cases} \mathcal{O}(n^2 \log n(\log\log r^* - \log\log(1+\varepsilon))) & , \textit{ whenever } r^* > 1+\varepsilon \\ \mathcal{O}(n^2 \log n(\log\log \frac{1}{r^*} - \log\log(1+\varepsilon))) & , \textit{ whenever } \frac{1}{r^*} > 1+\varepsilon \\ \mathcal{O}(n^2 \log n) & , \textit{ otherwise} \end{cases} ,$$

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

Additionally, it is useful to descend to a local solution after a new $r_{\max}$ is found. To this end, we store the line that reaches the highest level in DECIDELQD. It is possible to compute the lowest local solution on this line in time $\mathcal{O}(n^2 \log n)$. This descent method does not need much additional time and has the advantages that we may get faster over large gaps between local solutions and that we have a local solution that is an intersection of two lines (which increases the chance to reach the global solution).

4.2 Randomized Search

For the randomized search, we need initial values for $r_{\max}$ and $r_{\min}$. We set $r_{\min}$ to 0 and determine $r_{\max}$ by calculating a point on the $\binom{h}{2} + \binom{n}{2}$-level in the transformed input for a randomly chosen fixed u-coordinate in time $\mathcal{O}\left(n^2 \log n\right)$. 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. It is convenient to restrict the choice of the u-coordinate to intersections of the lines in modified dual space with the u-axis.
We denote the horizontal lines that correspond to $r_{\max}$ and $r_{\min}$ by $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. After determining $r_{\max}$ and $r_{\min}$ we calculate all intersections of the lines in modified dual space with the two horizontal lines $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. Assuming that no two lines in modified dual space intersect $\mathcal{H}(r_{\max})$ or $\mathcal{H}(r_{\min})$ in the same point, we sort the intersections on $\mathcal{H}(r_{\max})$ according to their horizontal position. Afterwards, we label the intersections from left to right with $\{1, \ldots, \binom{n}{2}\}$. If there are intersections in the same point, they get the same label. Intersection points on $\mathcal{H}(r_{\min})$ get the label ℓ, if they are located on the same line as the intersection point on $\mathcal{H}(r_{\max})$ labelled with ℓ. We now sort the intersections on $\mathcal{H}(r_{\min})$ according to their horizontal position and obtain a permutation π of $\{1, \ldots, \binom{n}{2}\}$. An inversion in a permutation is a pair of values where $i > j$ and $\pi(i) < \pi(j)$. An inversion table contains the number of inversions for each element. We additionally count inversions where $i < j$ and $\pi(i) > \pi(j)$. This inversion table uniquely determines how many intersections each line has between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$ and it can be computed in time $\mathcal{O}\left(n^2 \log n\right)$, for example with an extended merge sort algorithm. Note that the term 'between' excludes intersections on $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. We denote the set of intersections between to horizontal lines $\mathcal{H}(r_1)$ and $\mathcal{H}(r_2)$ by $\mathcal{I}(\mathcal{H}(r_1), \mathcal{H}(r_2))$.
Now, we randomly choose a number k between 1 and the number of intersections between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. We determine the "kth intersection" between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$ by finding that label ℓ where the sum of intersections of all smaller labels s is less than k and greater or equal to k if we add the intersections of ℓ between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. Afterwards, we determine the ith intersection on ℓ between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$ such that $s + i = k$ and denote the height of this intersection by r_{mid}. Afterwards, we use DECIDELQD(r_{mid}) and obtain a new value for $r_{\min}$ or $r_{\max}$ (depending on the decision of DECIDELQD) in time $\mathcal{O}\left(n^2 \log n\right)$.

Lemma 2. *The application of the above-described method assures*

$$\begin{aligned} \mathrm{E}(|\mathcal{I}(\mathcal{H}(r_{min}), \mathcal{H}(r_{mid}))|) &= \mathrm{E}(|\mathcal{I}(\mathcal{H}(r_{mid}), \mathcal{H}(r_{max}))|) \\ &\leq \frac{1}{2}(|\mathcal{I}(\mathcal{H}(r_{min}), \mathcal{H}(r_{max}))| - 1) \ . \end{aligned}$$

Proof. Let m be the number of intersections between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$. Each of these intersections is chosen with the same probability. Assume that no two intersections have the same v-coordinate. Then

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

Intersections with the same v-coordinate can only lead to a smaller value. □

Due to Lemma 2, we expect to have no intersection points between $\mathcal{H}(r_{\max})$ and $\mathcal{H}(r_{\min})$ after $\mathcal{O}(\log n)$ steps and therefore the optimal solution determined by

DECIDELQD($r_{\max}$). As each of these steps takes time $\mathcal{O}\left(n^2 \log n\right)$, we obtain the following result:

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

5 Experimental Results

While it is theoretically possible to choose ε in such a way, that the approximation algorithm is slower than the randomized algorithm, the approximative version is generally faster in practice. For the conducted experiments, we used 64 bit floating point numbers according to IEEE 754-1985. 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

$$\{(x_i, y_i) \mid x_i = \frac{2\,(i-1)}{n-1}; y_i = -x_i + 1.2 + e_1; 1 \leq i \leq n\}$$

and the second is

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

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. Below, 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.

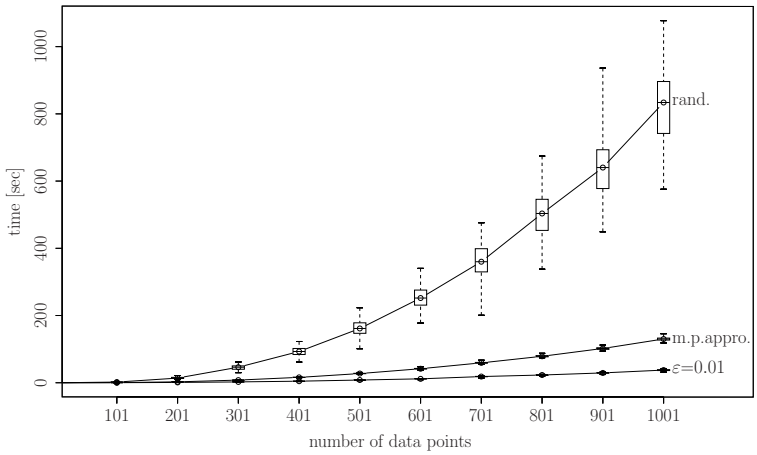

Fig. 3. Running time in seconds on a Pentium 4 CPU with 2, 56GHz and 1024MB of RAM for the first type of data set.

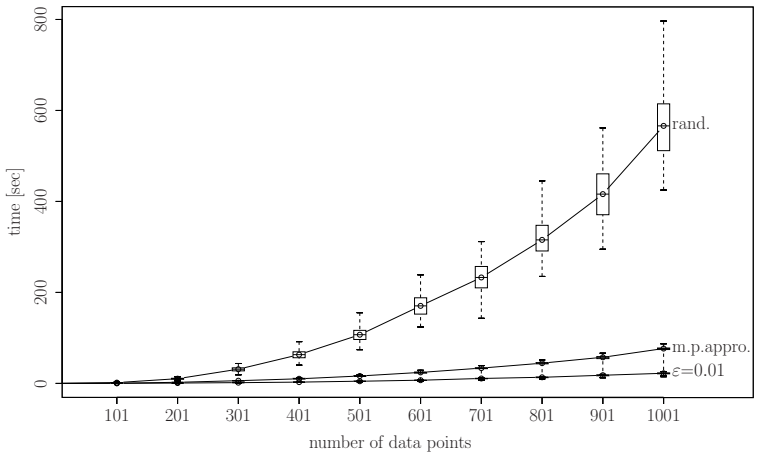

Fig. 4. Running time in seconds on a Pentium 4 CPU with 2, 56GHz and 1024MB of RAM for the second type of data set.

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 presented algorithms 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

1. Agulló, J.: An exchange algorithm for computing the least quartile difference estimator. Metrika **55** (2002) 3–16
2. Chazelle, B., Guibas, L.J., Lee, D.T.: The power of geometric duality. BIT **25** (1985) 76–90
3. Croux, C., Rousseeuw, P.J., Hössjer, O.: Generalized s-estimators. J. Amer. Statist. Assoc. **89** (1994) 1271–1281
4. Donoho, D., Huber, P.: The notion of breakdown point. In Bickel, P., Doksum, K., Hodges, J.J., eds.: A Festschrift for Erich L. Lehmann. Wadsworth (1983) 157–184
5. Dryden, I.L., Walker, G.: Highly resistant regression and object matching. Biometrics **55** (1999) 820–825
6. Edelsbrunner, H., Souvaine, D.: Computing least median of squares regression and guided topological sweep. J. Amer. Stat. Assoc. **85** (1990) 115–119
7. Langerman, S., Steiger, W.L.: The complexity of hyperplane depth in the plane. Discrete & Computational Geometry **30** (2003) 299–309
8. Mount, D.M., Netanyahu, N.S., Romanik, K., Silverman, R., Wu, A.Y.: A practical approximation algorithm for the LMS line estimator. In: SODA '97, SIAM (1997) 473–482
9. Rousseeuw, P.J.: Least median of squares regression. J. Amer. Statist. Assoc. **79** (1984) 871–880
10. Rousseeuw, P.J., Hubert, M.: Regression depth. J. Amer. Statist. Assoc. **94** (1999) 388–402
11. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. John Wiley & Sons Inc., New York (1987)
12. Rousseeuw, P., Van Driessen, K.: Computing LTS regression for large data sets. Estadística **54** (2002) 163–190
13. Wand, J.N.A., Sekhon, J.S., Mebane, Jr., W.R.: A comparative analysis of multinomial voting irregularities: Canada 2000. In: Proceedings of the American Statistical Society. (2001)