Computing the Least Median of Squares Estimator in Time $O\left(n^d\right)$

Thorsten Bernholt*
Lehrstuhl Informatik 2, Universität Dortmund, Germany

February 25, 2005

Abstract

In modern statistics, the robust estimation of parameters of a regression hyperplane is a central problem, i. e., an estimation that is not or only slightly affected by outliers in the data. In this paper we will consider the least median of squares (LMS) estimator. For n points in d dimensions we describe a randomized algorithm for LMS running in $O\left(n^d\right)$ time and $O(n)$ space, for d fixed, and in time $O\left(d^3 \cdot (2n)^d\right)$ and $O(dn)$ space, for arbitrary d.

Keywords: Robust statistics, computational geometry, least median of squares estimator.

1 Introduction

A general problem in statistics is the characterization of a set of points $\mathcal{P}$ by a straight line. One well-known method is the ordinary least squares regression line, which is the line that minimizes the sum of the squared vertical point-line distances. The parameters of such a regression line are computed by calculating some sums, see e.g. [1]. Suppose a single point is moved towards infinity. As the sums are taken over all points, this point will have a massive impact on the regression line, so that, for example, one single measurement error could result in a totally wrong regression line. This leads to the definition of the breakdown point. Donoho and Huber [6] define: "The *breakdown point* is, roughly, the smallest amount of contamination that may cause an estimator to take on arbitrarily large aberrant values".

To cope with this problem the basic idea is to ignore a fraction of the points and base the regression line on the remaining "good" points. To decide which are the

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

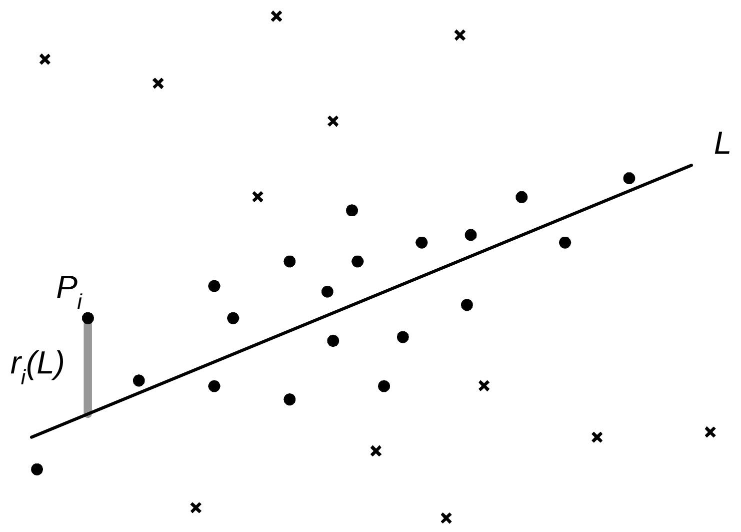

Figure 1: The statistical point of view (Definition 1.1) is displayed. The residual $r_i(L)$ is the vertical distance from the hyperplane L to the point P_i. The points marked with • have a residue less than r_i, the point marked with × have a larger residual. The solution to LMS is defined as the hyperplane such that the squared value of the h-th residual is minimized.

"good" points, each subset $S \subseteq \mathcal{P}$ is evaluated by a function $f : \mathbb{P}(\mathcal{P}) \longrightarrow \mathbb{R}^+$ and the subset with the best value is taken. $\mathbb{P}(\mathcal{P})$ denotes the set of all subsets of $\mathcal{P}$. Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of points with $P_i = (p_{i,1}, \ldots, p_{i,d})$ and $p_{i,j} \in \mathbb{R}$. For a given hyperplane L with the parameters $a_1, \ldots, a_d$ that is defined by

$$y = a_1 x_1 + \cdots + a_{d-1} x_{d-1} + a_d$$

let $r_i(L) = p_{i,d} - (a_1 p_{i,1} + \cdots + a_{d-1} p_{i,d-1} + a_d)$ be the residual of the point P_i with respect to the hyperplane L. A residual measures the vertical point-hyperplane distance. Let π be the permutation such that the sequence of the absolute values of all residues $|r_{\pi(1)}(L)|, \ldots, |r_{\pi(n)}(L)|$ is sorted. Hence, $r_{\pi(i)}(L)$ will denote the i-th residual in this order. Rousseeuw and Leroy [20] define the least median of squares estimators (LMS) (LMedS is also used as abbreviation in some papers) as follows:

Definition 1.1 (statistical point of view) *Given a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of n points, $P_i \in \mathbb{R}^d$ and a natural number h, with $\lceil n/2 \rceil \leq h \leq n$, find a hyperplane L, such that $r_{\pi(h)}(L)^2$ is minimized.*

The definition is illustrated in Figure 1. The highest breakdown point of 50% is achieved for $h = \lceil n/2 \rceil + \lceil (d+1)/2 \rceil$. One can choose $h = c \cdot n$ for a constant c with $0.5 \leq c \leq 1$, depending on the application and how many outliers are expected. This problem is also known as the least quantile of squares (LQS) estimator [21].

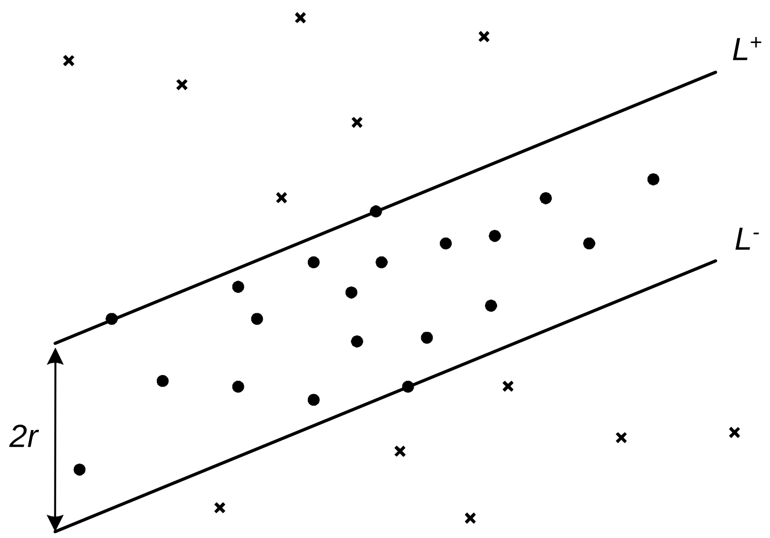

Figure 2: The computational point of view (Definition 1.2) is displayed. The hyperstrip defined by L^+ and L^- divides the points in inlying points • and outlying points ×. The half width of the hyperstrip is r. The solution for LMS is the hyperstrip with the smallest r containing h points.

For a moment, choose $r = |r_{\pi(h)}(L)|$. Now, consider the two hyperplanes

$$L^+ : y = a_1x_1 + \cdots + a_{d-1}x_{d-1} + a_d + r \tag{1}$$

$$L^- : y = a_1x_1 + \cdots + a_{d-1}x_{d-1} + a_d - r \ . \tag{2}$$

The two parallel hyperplanes L^+ and L^- forms a *hyperstrip*. All points with a residual smaller than or equal to r are inside this hyperstrip. That means, replacing the "=" with a "$\leq$" in Equation (1) and a "$\geq$" in Equation (2), these points will fulfill these inequations. The *width* of this hyperstrip is $2r$. Denote by $subset(L^+, L^-) \subseteq \mathcal{P}$ the set of points that are inside the hyperstrip (L^+, L^-). We now rephrase the definition slightly, whereby both definitions describe the same problem:

Definition 1.2 (computational point of view) *Let a set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of n points, $P_i \in \mathbb{R}^d$ and a natural number h, with $\lceil n/2 \rceil \leq h \leq n$, be given. Now find a hyperstrip (L^+, L^-) respectively its parameter $a_1, \ldots, a_d, r$, with $r \geq 0$, such that $|subset(L^+, L^-)| = h$ and the half width r of the hyperstrip is minimized.*

The definition is illustrated in Figure 2. There is a wide interest in this topic. Stromberg [21] gives an exact algorithm for LMS running in time $O\big(n^{d+2} \log n\big)$. Erickson et al. [10] describe an algorithm for LMS with running time $O\big(n^d \log n\big)$.We describe a randomized algorithm with a running time $O\big(n^d\big)$ for LMS for d fixed, as commonly used in the literature. All mentioned algorithms deal with intersection of hyperplanes and therefore have to solve systems of linear equations. A linear system can be solved in time $O(d^3)$. For arbitrary dimension d, the algorithm, described here, has a runtime of $O\big(d^3 \cdot (2n)^d\big)$ and needs space $O(dn)$.

For two dimensions, Edelsbrunner and Souvaine [9] describe an algorithm running in time $O(n^2)$ using the topological sweep-line technique [19]. Mount et al. [15] use branch-and-bound to compute LMS in the plane and simulations show a running time of $O(n \log n)$. This algorithm in addition computes approximate solutions. There are two ways to approximate LMS: The first way is to find a solution with fewer than h points inside (quantile approximation) and the second way is to find a solution with a wider hyperstrip (width approximation). For fixed dimension d, Olson [16] describes a width approximation algorithm with a ratio of 2 running in time $O\big(n^{d-1} \log n\big)$. Mount et al. [14] present a quantile approximation algorithm with a ratio $1-\epsilon$ and a running time of $O\big(n \log n + (1/\epsilon)^{O(d)}\big)$, also for fixed d. For h close to n, an algorithm of Chan [3] is useful, which uses linear programming with violations and solves LMS in time $O\big(n \log(n-h) + (n-h)^2 \log^2(n-h)\big)$.

The complexity of LMS is analyzed by Chien and Steiger [4]. They proved a lower bound of $\Omega(n \log n)$ in the model of algebraic decision trees. Gajentaan and Overmars [11] introduce the concept of 3-sum-hardness. Given n integer numbers, the 3-sum problem is to decide whether three distinct numbers sum up to zero. Besides reductions to other problems, they proved that, if one can solve LMS in $o(n^2)$, then the 3-sum problem, and others, can also be solved in $o(n^2)$. Erickson et al. [10] proof that if the affine degeneracy problem requires $\Theta(n^d)$ time, then the computation of width-LMS requires $\Omega(n^{d-1})$ time and of the exact LMS requires $\Omega(n^d)$ time, matching the running time of the algorithm presented in this paper, for d fixed.

The estimator is widley used, e. g., Plets and Vynckier [18] use the LMS-estimator to analyse 3-dimensional astronomical data. In [17] the estimator is used for mosaicing underwater images, moreover in [13] images of a mpeg stream are used to compose a panorama view.

In Section 2 we describe the main procedure of the algorithm. It uses procedures from Sections 3, 4 and 5.

2 Main Procedure

In this section, we give a short overview, how the algorithm works. For $d = 2$, we use the algorithm of Edelsbrunner and Souvaine [9] for LMS in the plane working in time $O(n^2)$ and space $O(n)$. For $d \geq 3$, at first, the d-dimensional points are mapped to $2n$ hyperplanes in a $d+1$-dimensional space. The details are given in Section 3. In this space, a solution is represented by a point. An optimal solution is a point L' in $\mathbb{R}^{d+1}$ with the following three properties (Lemma 3.2):

1. There are $n + h$ hyperplanes below or intersecting the point L'.
2. The last coordinate (w-axis) of the point L' is minimal over all points with Property 1.
3. The point L' is an intersection of at least $d + 1$ hyperplanes.

For two selected hyperplanes a *subproblem* consists of all points contained in the subspace described by these two hyperplanes. Therefore, we obtain $\binom{n}{2}$ subproblems $A_1, \ldots, A_{\binom{n}{2}}$. It is essential that the subproblems are permuted, i. e., they are in a random order. The optimal solution r^* of the first subproblem A_1 is computed using procedure SolveLMS, which has a runtime of $O\big(d^3 \cdot (2n)^{d-1}\big)$. The details of the procedure SolveLMS are given in Section 4.

Consider that we have already computed the optimal solution of $A_1, \ldots, A_{i-1}$. For the succeeding subproblem A_i it is decided in time $O\big(d^3 \cdot (2n)^{d-2}\big)$ whether in this subproblem a better solution than r^* exists. This is done by the application of the procedure DecideLMS, the details are given in Section 5. If a better solution exists, it is computed using SolveLMS, r^* is updated, and the algorithm continues with the next subproblem A_{i+1}, until all $\binom{n}{2}$ subproblems are processed. The time for all calls of DecideLMS are bounded by $O\big(d^3 \cdot (2n)^{d}\big)$.

To get the runtime for the calls of SolveLMS, we name the event, that r^* is improved, Ψ. As the subproblems are in a random order, the event Ψ only occurs $\log \binom{n}{2} = O(\log(n))$ times in the average case. This is a well-known result from [5] and also, e. g., discussed by Chan [2]. Therefore, in the average case the time for all calls of SolveLMS is bounded by $O\big(d^3 \cdot (2n)^{d-1} \cdot \log n\big)$.

For d=3, we have to reduce the number of calls to DecideLMS. Therefore, two levels of calls are needed. The first level processes subproblems defined by a single hyperplane and each call to DecideLMS can be computed in time $O(n^2)$. The second level remains as described above and it is called only $O(\log n)$ times in the average case. Hence, the number of calls of DecideLMS is reduced to $O(n \log n)$, and we get a runtime of $O(n^3)$. This proves the following theorem:

Theorem 2.1 *Given n points in $d \geq 2$ dimensional space, d not fixed, the LMS estimator can be computed in expected time $O\big(d^3 \cdot (2n)^d\big)$ and space $O(dn)$.*

A result from Karp [12] shows, that the probability, that the event Ψ occurs more than $2n$ times, is bounded by $\left(\frac{1}{2}\right)^n$. But, due to the lack of space, this is not explained in detail here.

3 Transformation of the Input

According to the point-hyperplane duality, we map a point $P_i = (p_{i,1}, \ldots, p_{i,d})$ from primal space to the hyperplane H_i defined by

$$v = p_{i,1}u_1 + \cdots + p_{i,d-1}u_{d-1} + p_{i,d}$$

In a second step we map from dual space to extended space with the axes $u_1, \ldots,$ u_{d-1}, v, w and map the hyperplane H_i to the two hyperplanes

$$H_i^+ : w = +p_{i,1}u_1 + \cdots + p_{i,d-1}u_{d-1} + v - p_{i,d} \tag{3}$$
$$H_i^- : w = -p_{i,1}u_1 - \cdots - p_{i,d-1}u_{d-1} - v + p_{i,d} \ . \tag{4}$$

In the extended space, we say that the point $(q_1, \dots, q_d, q_{d+1})$ is located *above* the hyperplane H if there exists a constant $c > 0$ such that the point $(q_1, \dots, q_d, q_{d+1} - c)$ is located on the hyperplane H. The terms *below* is defined analogously.

What happens to a solution during the mapping? In the primal space, a solution consists of two parallel hyperplanes with h points between them. This is mapped to a vertical segment with h hyperplanes crossing it. Finally, the vertical segment is mapped to a point in $(d+1)$-dimensional space with $n+h$ hyperplanes below it. More precisely:

Lemma 3.1 *Let $L = (L^+, L^-)$ be a solution with value $r \geq 0$ in the primal space with*

$$
\begin{aligned}
L^+ &: y = a_1 x_1 + \dots + a_{d-1} x_{d-1} + a_d + r \\
&\textit{and} \\
L^- &: y = a_1 x_1 + \dots + a_{d-1} x_{d-1} + a_d - r \ .
\end{aligned}
$$

Let L' be the solution in the extended space defined by the point $(a_1, \dots, a_d, r)$. Now the point P_i is located between L^+ and L^- if and only if L' is located above or on H_i^+ and above or on H_i^-.

Proof: L' is above or on H_i^+

$$
\begin{aligned}
&\Leftrightarrow \ \exists c \geq 0 : (a_1, \dots, a_d, r - c) \text{ is located on } H_i^+ \\
&\Leftrightarrow \ \exists c \geq 0 : r - c = p_{i,1} a_1 + \dots + p_{i,d-1} a_{d-1} + a_d - p_{i,d} \\
&\Leftrightarrow \ \exists c \geq 0 : p_{i,d} - c = a_1 p_{i,1} + \dots + a_{d-1} p_{i,d-1} + a_d - r \\
&\Leftrightarrow \ \exists c \geq 0 : (p_{i,1}, \dots, p_{i,d-1}, p_{i,d} - c) \text{ is located on } L^- \\
&\Leftrightarrow \ P_i \text{ is above or on } L^-
\end{aligned}
$$

An analogous calculation shows that L' is above or on H_i^- if and only if P_i is below or on L^+. □

An optimal solution is characterised by the following lemma:

Lemma 3.2 *An optimal solution for LMS in the extended space is a point L' with the following properties:*

1. *There are $n + h$ hyperplanes below or intersecting the point L'.*
2. *The last coordinate of the point (w-axis) is minimal over all points with Property 1.*
3. *The point is an intersection of at least $d + 1$ hyperplanes.*

Proof:

1. Let the hyperstrip (L^+, L^-) be an optimal solution for LMS. Then there are h points that are located between L^+ and L^-. It follows from Lemma 3.1 that there are h pairs (H_i^+, H_i^-), such that both hyperplanes are below the point L'. As the w-coordinate of the point L' is $r \geq 0$, it follows from the Equations (3) and (4) that for all $i = 1, \ldots, n$ either a hyperplane H_i^+ or H_i^- is below the point. Therefore, there are $n + h$ hyperplanes below the optimal solution L'.

2. As stated in Definition 1.2 the solution of LMS is a hyperstrip of minimal width $2r$. As the half width of a hyperstrip and the w-coordinate of the dual point are equal to r, the solution is a point with a minimal w-coordinate with respect to Property 1.

3. If there is a point that is an intersection of less than $d+1$ hyperplanes, we can find a point with a smaller w-coordinate that is an intersection of at least $d+1$ hyperplanes. Thus, a point that is intersected by less than $d+1$ hyperplanes cannot be optimal.

□

4 Procedure SolveLMS

The main algorithm has fixed two hyperplanes that define a lower-dimensional subspace and therefore define a subproblem A. To enumerate all solutions contained in the subspace of A, we loop through all subdivisions that are described by fixing $d-3$ hyperplanes in addition. This results in $\binom{2n-4}{d-3} \leq (2n)^{d-3}$ many subdivisions. In [7] an algorithm for enumerating all $(d-1)$-elementary subsets is given with a constant runtime per subset. If the hyperplanes, that define a subdivision, are linear independent, they describe a two-dimensional subspace. Otherwise, they describe a subspace with more than two dimensions, and such a subdivision can be ignored, as it is easy to see, that such a subdivision is covered by other linear independent subdivisions.

Each two-dimensional subspace is processed in the following way: The intersection of such a two-dimensional subspace with the remaining $n' = 2n - (d-1)$ hyperplanes results in a set of n' lines embedded in $\mathbb{R}^{d+1}$. Defining two axes in the two-dimensional subspace results in n' lines in the plane. We can now use the sweep-line algorithm from [8, 19] to enumerate all $\binom{n'}{2}$ intersection points. This algorithm works in time $O(n'^2)$ and space $O(n')$. For each intersection point we check if in the $d+1$-dimensional space exactly $n+h$ hyperplanes are below the point or intersecting it. To make this routine work in time $O(n'^2)$, we track a count for each line. The count for the first intersection point found on a line can be calculated in $O(n')$. The counts of the succeeding intersection points can be calculated by an update step in time $O(1)$, as the sweep-line algorithm reports the points in a topological order.

The required space is $O(n')$ overall. The point with $n+h$ hyperplanes below and with the smallest w-coordinate is taken as the new optimum.

The procedure deals with $O\big((2n)^{d-1}\big)$ intersection points. For each point we have to compute the intersection of up to $d+1$ hyperplanes. Each hyperplane is indeed an equation, therefore we have to solve a system of up to $d+1$ equations over $d+1$ variables. For d not fixed, this task can be performed in time $O(d^3)$, using, for example, Gaussian-elimination or other numerical algorithms. As a single point/hyperplane has $d+1$ entries, $O(dn)$ space is sufficient. Therefore, the procedure SolveLMS finds the optimal solution of a subproblem A in time $O\big(d^3 \cdot (2n)^{d-1}\big)$ and space $O(dn)$ for $d \geq 3$.

5 Procedure DecideLMS

Given a subproblem A and a value r^*, we want to decide whether there exists a solution for LMS with a value better than r^* in A. Recall that the subproblem A is a $(d-1)$-dimensional subspace embedded in $\mathbb{R}^{d+1}$. This only holds if the two defining hyperplanes are linearly independent. Otherwise, the subproblem can be ignored, as discussed in Section 4.

We know from Lemma 3.2 that the w-axis measures the value of a solution. Therefore, we intersect this subspace with the hyperplane $H_{r*} : w = r^*$, resulting in a $(d-2)$-dimensional subspace. All points in this subspace represent solutions having the same value r^*, the question is whether there is a point with $n+h$ hyperplanes below it.

Lemma 5.1, which is presented below, shows that it is sufficient to enumerate all $\binom{2n-4}{d-2} \leq (2n)^{d-2}$ intersection points of the remaining hyperplanes with the subspace $A \cap H_{r*}$ and count the number of hyperplanes below. To enumerate all intersection points we use the same technique as in Section 4. Therefore, Procedure DecideLMS computes the decision, whether a better solution exists, in time $O\big(d^3 \cdot (2n)^{d-2}\big)$ for $d \geq 3$. For calls to this procedure on the second level ($d=3$) only a runtime of $O(n \log n)$ can be achieved since the considered subspace is one-dimensional and we have to sort the intersection points.

Lemma 5.1 *If the subproblem A contains a solution for LMS with a value smaller than r^*, then the intersection of A and H_{r*} contains an intersection point L' such that at least $n+h$ hyperplanes are below L'.*

Proof: In general, the set of $d+1$ hyperplanes intersecting an optimal solution either contains two H^+- or contains two H^--hyperplanes. W.l.o.g. we focus on the case that the subproblem A is an intersection of the hyperplanes H_i^+ and H_j^+, as displayed in Figure 3. Let $L'' = (a_1, \ldots, a_{d-1}, v, \lambda)$ be the optimal solution in A. As the optimal solution L'' is located in A, it intersects the two hyperplanes and we get

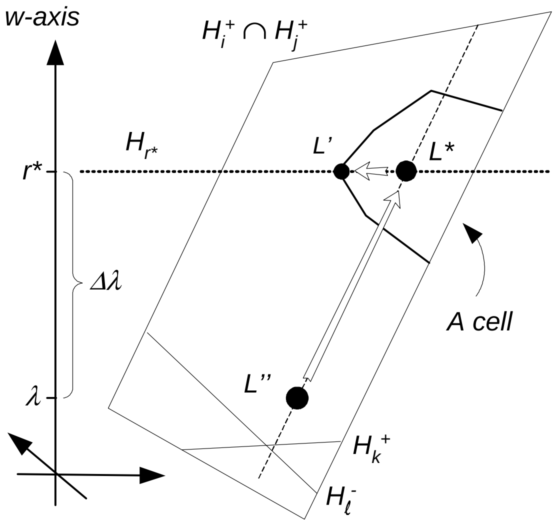

Figure 3: The point L'' is moved towards L^*, the hyperplanes H_k^+ and H_ℓ^- remain below. The point is further moved towards L' in the corner of a cell, now intersecting $d+1$ hyperplanes.

from Equation (3)

$$\lambda = p_{i,1}a_1 + \cdots + p_{i,d-1}a_{d-1} + v - p_{i,d}$$
$$\lambda = p_{j,1}a_1 + \cdots + p_{j,d-1}a_{d-1} + v - p_{j,d} \quad .$$

For clarity, we combine most of the terms into Q_i and Q_j and we get

$$\lambda = Q_i + v$$
$$\lambda = Q_j + v \quad .$$

If we increase λ by $\Delta\lambda > 0$, we can reason that the point $L^* = (a_1, \ldots, a_{d-1}, v + \Delta\lambda, \lambda + \Delta\lambda)$ still intersects the two hyperplanes. We choose $\Delta\lambda$ such that $\lambda + \Delta\lambda = r^*$.

Let the hyperplanes H_k^+ and H_ℓ^- be located below L''. Then the point L'' satisfies the following two inequalities:

$$\lambda \geq Q_k + v$$
$$\lambda \geq -Q_\ell - v$$

A simple calculation shows, that L^* also satisfies the inequalities. Therefore, all hyperplanes located below L'' are also located below L^*. Since $\lambda > 0$, hyperplanes intersecting L'' are now below L^*. Well, there maybe some hyperplanes below L^* in addition. Hence, there are at least $n + h$ hyperplanes below L^*. If the point L^* is not an intersection of $d+1$ hyperplanes, we move the point L^* towards a nearby intersection point L', such that it does not leave the subspace $A \cap H_{r*}$ and such that no hyperplane is crossed. □

6 Conclusions

In this paper, we have described a randomized algorithm for the least median of squares estimator for n points in d dimensions. It runs in time $O\big(d^3 \cdot (2n)^d\big)$ and space $O(dn)$. For d fixed, the runtime matches the lower bound of $\Omega(n^d)$.

Acknowledgement

I would like to thank Ingo Wegener for helpful suggestions concerning the presentation of the results and Claudia Becker, Thomas Fender and Ursula Gather for introducing me to the questions of robust statistics.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[2] T. M. Chan. Geometric applications of a randomized optimization technique. *Discrete & Computational Geometry*, 22:547–567, 1999.

[3] T. M. Chan. Low-dimensional linear programming with violations. In *IEEE Symposium on Foundations of Computer Science*, pages 570–579, 2002.

[4] H. Chien and W. Steiger. Some geometric lower bounds. In *ISAAC: 6th International Symposium on Algorithms and Computation*, pages 72–81, 1995.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition).* MIT Press & McGraw-Hill, 2001.

[6] D. L. Donoho and J. Huber. The notion of breakdown point. In P. Bickel, K. Doksum, and J. L. Hodges, editors, *A Festschrift for Erich L. Lehmann*, pages 157–184, 1985.

[7] P. Eades and B. McKay. An algorithm for generating subsets of fixed size with a strong minimal change property. *Information Processing Letters*, 19:131–133, 1984.

[8] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *Journal Computer and System Sciences*, 38:165–194, 1989, Corrigendum in 42:249–251,1991.

[9] H. Edelsbrunner and D. L. Souvaine. Computing least median of squares regression line and guided topological sweep. *Journal of the American Statistical Association*, 85:115–119, 1990.

[10] J. Erickson, S. Har-Peled, and D. M. Mount. On the least median square problem. In *ACM Symposium on Computational Geometry*, 2004.

[11] A. Gajentaan and M. H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry*, 5:165–185, 1995.

[12] R. M. Karp. Probabilistic recurrence relations. *Journal of the Association of Computer Machinery*, 41(6):1136–1150, 1994.

[13] Y. Li, L-Q. Xu, G. Morrison, C. Nightingale, and J. Morphett. Robust panorama from mpeg video. In *IEEE International Conference on Multimedia and Expo 2003 (ICME '03)*, pages 81–84, 2003.

[14] D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. Quantile approximation for robust statistical estimation and k-enclosing problems. *International Journal of Computational Geometry and Applications*, 10:593–608, 2000.

[15] D. M. Mount, N. S. Netanyahu, K. Romanik, R. Silverman, and A. Y. Wu. A practical approximation algorithm for the LMS line estimator. In *Symposium on Discrete Algorithms*, pages 473–482, 1997.

[16] C. F. Olson. An approximation algorithm for least median of squares regression. *Information Processing Letters*, 63(5):237–241, 1997.

[17] O. Pizarro and H. Singh. Toward large-area mosaicing for underwater scientific applications. *IEEE Journal of Oceanic Engineering*, 28(4):651–672, 2003.

[18] H. Plets and C. Vynckier. An analysis of the incidence of the vega phenomenon among main-sequence and post main-sequence stars. *Astronomy and Astrophysics*, 343:496–506, 1999.

[19] E. Rafalin, D. Souvaine, and I. Streinu. Topological sweep in degenerate cases. In *International Workshop on Algorithm Engineering and Experiments*, pages 155–165, 2002.

[20] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[21] A. J. Stromberg. Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression. *SIAM Journal on Scientific Computing*, 14(6):1289–1299, 1993.