ARRANGEMENT SEARCHING AND APPLICATIONS TO ROBUST REGRESSION

J. Michael Steele, Princeton University

1. Introduction

The purpose of this report is to show the applicability of some recent work in computer science to a problem of current interest in robust regression. In particular, point-line duality and line-sweep methods are applied to provide an algorithm for computing least median of squares regression lines. The algorithm which is obtained has a worst case time complexity of $0(n^2 \log n)$ using only $0(n)$ space. This report should be thought of as an introduction to the more detailed papers Steele and Steiger (1986) and Souvaine and Steele (1986).

First, there is a brief review of some notions of robust regression and the motivation for regression methods with high breakdown point. The least median of squares regression method is introduced, and its computational issues are reviewed. The third section discusses point-line duality and its application to the problem of least median of squares regression. The fourth section develops the key observations of the sweep line method and sketches an algorithm for least median of squares regression.

The final section provides information about a second algorithm which is detailed in Souvaine and Steele (1986). The second algorithm uses more sophisticated data structures, has worst case complexity of only $0(n^2)$, but suffers from a $0(n^2)$ space requirement. Additionally, the final section reviews some other recent work in computer science which can be expected to have an impact on work in computational statistics.

2. Notions of Robust Regression and Breakdown Point

The breakdown point of an estimator is (roughly) the smallest amount of contamination of the data which can cause the estimator to take on arbitrary large values. The first extensive discussion of breakdown point is given in Donoho and Huber (1983), but they trace the concept to the thesis of F.R. Hample (1968). The discussion of Donoho and Huber lends much credence to the notion that the crude notion of breakdown is a very useful measure of robustness, especially as applied in small sample contexts.

Information about the breakdown properties of many robust regression proposals is provided in the key paper by Rousseeuw (1984) on Least Median of Squares Regression (LMS regression), and LMS regression if found to compare favorably with all current proposals.

Formally, if we have data (x_i, y_i), $1 \le i \le n$, and we wish to fit a line $y = ax + b$, we consider the set of residuals $r_i = y_i - ax_i - b$ associated with a and b. We then choose as estimates the $\hat{a}$ and $\hat{b}$ which minimize the feature of merit

$$f(a,b) = \underset{1 \le i \le n}{\text{median}}\ r_i^2.$$

One point to note is that if *median* is replaced by mean then the corresponding feature of merit would lead to a rephrasing of the procedure of ordinary least squares.

The first considerations of the computation of LMS regression are in Rousseeuw (1984) and Leroy and Rousseeuw (1984), but the first formal study was given in Steele and Steiger (1986). For (x_i, y_i) in general position, it was proved that $f(\alpha, \beta)$ has at least cn^2 local minima. One consequence of this large number of local minima is that traditional local optimization methods will prove fruitless for determining $\hat{a}$ and $\hat{b}$.

One must confront a combinatorial problem, and Steele and Steiger (1986) showed that there is a reduction to a tractable combinatorial procedure which yields an exact minimization of f via a finite search with worst time complexity of $0(n^3)$.

Souvaine and Steele (1986) give two faster algorithms for computing LMS regressions. The first of these depends upon the technique of point line duality and the sweep-line method. This algorithm was shown to have worst case time complexity $0(n^2 \log n)$ with a space requirement of $0(n)$. The second algorithm required more complex data structures, needed space of size $0(n^2)$, but was able to provide worst case time complexity of $0(n^2)$. For the purpose of this exposition, perhaps the best objective is to review just the first of these two methods and suggest how such technology is used in LMS regression. Suggestions are also made that might be useful in related work.

3. Point Line Duality and Its Application

The duality which will concern us here is given explicitly by the transformation T on points and lines which takes the point (a,b) to the line $y = ax + b$ and which takes the line $y = ax + b$ to the point $(-a,b)$.

Under this transformation, the point P that is determined by two lines L_1 and L_2 is mapped to the line TP which is determined by the two points TL_1 and TL_2. Likewise the line L determined by two points P_1 and P_2 is mapped to the point TL determined by the two lines TP_1 and TP_2. These relations lie at the base of duality and they are shared by the classical transformation S of Poncelet which maps the point (a,b) to the line $ax + by + 1 = 0$ and maps the line $ax + by + 1 = 0$ to the point (a,b). It is interesting to note that $S^2 = I$, but $T^2 \neq I$ where I is the identity mapping. This small loss of mathematical symmetry is well compensated by other properties of T which will be identified shortly.

In statistical work the point/line duality T has been used most recently in the work of Johnstone and Velleman (1985) which gives an algorithm for a different robust regression method, the resistant line. In data analysis and in simple linear regression, duality has also been used by Emerson and Hoaglin (1983), Dolby (1960), and Daniels (1954).

The benefit of the transformation T is that, in addition to the basic duality properties, it has some important order invariance properties. The first of these ordering properties is that the transformation T preserves the relationship of

"above" and "below" for points and lines. That is, if P is above L then TP is above TL. Secondly, the transformation T preserves the vertical distance between points and lines; TL is above TP by exactly the same distance that L is above P, *etc.* Understandably, this property is crucial in a problem where a median residual is to be minimized over a selection of lines. One simply has to have some form of isometric invariance in the duality if it is to prove at all useful.

The final important order invariance property of T is that if the slope of L is larger than the slope of L', then the x-coordinate of TL is smaller than the x-coordinate of TL'. This relation is trivial from the definition, but it is still very useful in practical applications.

In order to phrase LMS regression as a finite problem which can be precisely dualized, it is worthwhile to introduce some terminology relevant to its local structure. First, for any α and β, the line $l_{\alpha,\beta} = \{(x,y)\colon y = \alpha x + \beta\}$ defines residuals $r_i(\alpha,\beta)$ which are functions of α and β. We can typically write $r_i(\alpha,\beta)$ as r_i without fear of confusion. We say the line $l_{\alpha,\beta}$ *bisects* three distinct points (x_{i_j},y_{i_j}) $j = 1,2,3$ if all of the r_{i_j} are of the same magnitude r but not all have the same sign. If $x_{i_1} < x_{i_2} < x_{i_3}$ and $r_{i_1} = -r_{i_2} = r_{i_3}$ we say $l_{\alpha,\beta}$ *equioscillates* with respect to the points.

It was proved in Steele and Steiger (1986) that the LMS regression line must be an equioscillating line. Since there are at most $\begin{bmatrix} n \\ 3 \end{bmatrix}$ such lines, a naive algorithm would be to examine them all. Obviously, any algorithm faster than $0(n^3)$ must work differently.

Given an equioscillating line $l_{\alpha,\beta}$ there are two related lines with slope α; the line L_1 determined by the points $P_1 = (x_{i_1},y_{i_1})$ and $P_3 = (x_{i_3},y_{i_3})$, and the line L_2 which goes through the point $P_2 = (x_{i_2},y_{i_2})$. A key property of the LMS regression line $l_{\alpha,\beta}$ is that the number K of points between L_1 and L_2 must satisfy

$$K = \begin{cases} (n-4)/2 & n \text{ even} \\ (n-5)/2 & n \text{ odd} \end{cases}$$

provided $n > 3$ (c.f. Steele and Steiger (1986), Main Lemma).

One simple method of determining the LMS regression line can now be given. For each triple of data points P_1, P_2, and P_3 which are ordered by x-coordinate, we first determine lines L_1 and L_2 as above. Next we find such a triple with exactly K data points between L_1 and L_2 and such that the vertical distance between L_1 and L_2 is minimized. The LMS regression line is the line which goes exactly between L_1 and L_2.

We can now see the rephrasing of our problem obtained by applying T to the data points and the lines they determine. The condition that there are K points between the lines L_1 and L_2 becomes the condition that there are K lines between the points TL_1 and TL_2. Further, TL_1 is the point determined by the two lines TP_1 and TP_3; TL_2 is the point on the line TP_2 which has the same x-coordinate as the point TL_1.

Our algorithmic problem in dual can now be expressed as follows:

> Given n lines L_i, $1 \le i \le n$, in general position in the plane with intersection points P_{ij}, $1 \le i < j \le n$, find that line L^* and intersection point P^* such that among all line-point pairs (L,P) which have exactly K of the L_i cutting the vertical segment S joining L and P, the pair (L^*,P^*) has the smallest vertical distance.

The sweep-line technique for solving this dualized problem will be given in the next section.

4. Sweep-Line Algorithm and Space Efficiency

The sweep-line technique "sweeps" the plane by combinatorially moving a vertical "sweep-line" from the left to the right. Souvaine and Steele (1986) provide what is probably the first application of this technique in the area of regression, but Shamos and Hoey (1976) and Bentley and Ottman (1979) have shown that sweep-line techniques are applicable to a variety of problems in computational geometry.

The sweep-line algorithm requires two off-the-shelf data structures. The first of these (to be called) LIST will maintain our set of n lines in a geometrically meaningful order. LIST can be implemented as a simple linked list but we will want to augment it with some additional pointers which will help LIST interact with the second data structure.

Our second structure will be used to store a certain subset of $n-1$ out of the set of $n(n-1)/2$ intersection points P_{ij}. The structure we create will permit us to access the "smallest" element in time 0(1) and permits the insertion or deletion in time $0(\log n)$. The ordering we will use to give meaning to "smallest" is that $P_{ij} \ll P_{st}$ provided the x-coordinate of P_{ij} is smaller than the x-coordinate of P_{st}. To implement this second structure we can use a heap (see *e.g.* Aho, Hopcroft, and Ullman (1974)). We will refer to our particular structure as HEAP.

The initialization of these structures will show how they will be used and will help make the algorithm transparent. We begin by computing a value A such that all of the intersection points lie to the right of a line $L_Q\colon x = A$. This would be trivial to do this task in time $0(n^2)$ by computing the values P_{ij} successively and retaining only the left-most. It is instructive (and illustrative of a basic technique) to see how this can be done in time $0(n \log n)$.

To find A fast, we first note that to the left of all of the intersection points the lines are themselves ordered by slope, *i.e.* the line with lowest slope is above all of the other lines, *etc.* This observation implies that if we first sort the lines L_i, $1 \le i \le n$, by slope then the left-most intersection point P_{ij} must be an intersection of two lines which are adjacent in the list of lines sorted by slope. Thus, to find the left-most intersection point we can successively compute the intersection points of adjacent elements of the slope-sorted list of lines, keep only the left-most P_{ij}, and proceed down the list. This computation requires

$O(n \log n)$ time because of the preliminary sorting of the lines by slope. We did not have to dig out a clever way of executing this step, but doing so sets up a similar idea which will be crucial for our overall time complexity.

We next introduce another vertical line L which we call the sweep-line and we initialize L to L_o. The line L is not essential to the algorithm but it helps articulate a program invariant which helps make verification of the algorithm clearer.

We initialize LIST by placing the L_i into the list in decreasing order of L_o-intercept. We will also augment the structure holding the L_i by storing along with each of the L_i a set of four pointers Up1, UpK, Down1, and Down$K$ which will point to the lines which are 1 above L_i, K above L_i, and so forth in the ordering which is given by LIST. Since lines too near the top or bottom will not have lines satisfying the required conditions we set the pointers to the null value to flag this situation.

A key geometric observation parallel to the idea used to compute L_o is that from the whole set of $n(n-1)/2$ intersection points P_{ij}, $1 \le i < j \le n$, the one which is nearest to the sweep-line L must be determined by a pair of lines $\{L_i, L_j\}$ which are *adjacent* in an ordered list which was initialized by L_o-intercept. This observation permits us to keep a set of candidates for the "next point to explore" which is at most of cardinality $n-1$.

Formally, we pass through LIST and for each adjacent pair L_i and L_j in LIST we install P_{ij} in HEAP. We do this while respecting the ordering of the P_{ij} by x-coordinate and we end up with a heap of size $n-1$ with the left-most of the P_{ij} installed at the root. As an important adjunct to the heap building process we make a double set of pointers which associate the elements of HEAP and the elements of LIST. Specifically, for each adjacent pair L_i and L_j in LIST (and each corresponding intersection point P_{ij} in HEAP) we create pointers from L_i and L_j to P_{ij} and pointers from P_{ij} to L_i and L_j.

This step completes our preprocessing and initialization phase, so we should make a preliminary tally of resources consumed. First, because of the sorting, LIST requires time $O(n \log n)$ to build and space $O(n)$ to store. Since HEAP is also of size $O(n)$, it can be created in time $O(n)$ and space $O(n)$ using well-known methods.

The algorithm for computing the LMS regression line is now almost trivial using the data structures we have built. The picture to keep in mind is that of sweeping the vertical line L from L_o until it has passed through all of the points. What we will do is manage LIST and HEAP in such a way that (1) LIST always provides the ordering of L_i according to decreasing order of L-intercept, (2) the pointers stored in LIST remain consistent with their initial definition, and (3) HEAP always stores as its minimal element the intersection point P_{ij} which has smallest x-coordinate of any intersection point to the right of L.

5. Final Remarks

The sweep-line method which was just discussed is easy to implement and has substantial generality. Still, it is worth understanding that there are more sophisticated methods for the systematic search of an arrangement and there may be instances where the associated overheads (both computational and intellectual) are justified.

There is one particularly relevant data structure due to Chazelle (1984) called a *Hammock* and which is used in Souvaine and Steele (1986) to provide a $O(n^2)$ time $O(n^2)$ space algorithm for LMS regression. Moreover, there are data structures for searching higher dimensional arrangements due to Edelsbrunner *et.al.* (1983) which should be very useful in the study of LMS methods for multivariate regression (at least as a theoretical tool).

Finally, the technique of line sweep has been extended recently by Edelsbrunner and Guibas (1986) to what they call "topological sweeping of an arrangement." This is an extremely promising technology which will be sure to have an impact on computational statistics.

References

Aho, A.V., Hopcroft, J.E., and Ullman, J.D. (1974). *The Design and Analysis of Algorithms*, Adison-Wesley, Reading, MA.

Bentley, J.L. and Ottman, T.A. (1979). "Algorithms for reporting and counting geometric intersections", IEEE Trans. Comp., **C-28,** 643-647.

Chazelle, B. (1984). "Intersecting is easier than sorting", Proc. 16th ACM STOC, 125-135.

Chazelle, B. Guibas, L.J. and Lee, D.T. (1983). "The power of geometric duality," Proc. 24th IEEE FOCS, 217-225.

Daniels, H.E. (1954). "A distribution-free test for regression parameters," *Annals of Mathematical Statistics,* **25,** 499-513.

Dobkin, D.P. and Souvaine, D.L. (1986). "Computational Geometry - A Users Guide," *Advances in Robotics I: Algorithmic and Geometric Aspects of Robotics,* J.T. Schwartz and C.K. Yap (Eds.), Lawrence Erlbaum Associates, Hillsdale, NJ.

Dolby, J.L. (1960). "Graphical procedures for fitting the best line to a set of points," *Technometrics,* **2,** 477-481.

Donoho, D.L. and Huber P.J. (1983). "The notion of breakdown point", *A Festschrift for Erich L. Lehman,* P.J. Birkel, K. Doksum, and J.L. Hodges (Eds.), Wadsworth, Behmont CA, 157-184.

Edelsbrunner, H. O'Rourke, J. and Seidel, R. (1983). "Constructing arrangements of lines and hyperplanes with applications," Proc 24th IEEE FOCS, 83-91.

Edelsbrunner, H. and Guibas, L.G. (1986). "Topologically sweeping an arrangement," Technical Report, Stanford University, Department of Computer Science.

Emerson, J.D. and Hoaglin, D.C. (1983). "Resistant lines for y-versus-x." In D.C. Hoaglin, F. Mosteller and J. Tukey (Eds.), *Understanding Robust and Exploratory Data Analysis,* John Wiley, New York.

Hample, F.R. (1968). "Contributions to the Theory of Robust Estimation," Ph.D. Thesis, University of California, Berkeley.

Harary, Frank (1972). *Graph Theory*, Addison-Wesley Publishing Company, Menlo Park, CA.

Johnstone, I.M. and Velleman, P.F. (1985). "The resistant line and related regression methods," *J. Amer. Statis. Assoc.,* **80,** 1041-1054.

Leroy, A. and Rousseeuw, P.J. (1984). "PROGRESS: A program for robust regression," Report 201, Centrum voor Statistiek en Operationell Onderzoek, Univ. Brussels.

Rousseeuw P.J. (1984). "Least median of squares regression," *J. Amer. Statist. Assoc.,* **79,** 871-880.

Shamos, M. and Hoey, D. (1976). "Geometric intersection problems, IEEE FOCS Conference, Houston, TX.

Souvaine, D.L. and Steele, J.M. (1986). "Time and space efficient algorithms for least median of squares regression," Technical Report, Program in Statistics and Operations Research, Princeton University.

Steele, J.M. and Steiger, W.L. (1986). "Algorithms and complexity for least median of squares regression," *Discrete Applied Mathematics,* **13** 509-517.