November, 1984

ALGORITHMS AND COMPLEXITY FOR A STATISTICAL PROBLEM. MINIMUM MEDIAN RESIDUAL FITTING

J.M. Steele* and W.L. Steiger**

DCS-TR-149

*Department of Statistics
Princeton University

**Department of Computer Science
Rutgers University

Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

ABSTRACT

Given n points $\{(x_i,y_i)\}$ in the plane we study the problem of fitting the minimum median squared residual (**MM^{2}R**) line. This involves the study of the function $f(\alpha,\beta) = \text{median}(|y_i - (\alpha + \beta x_i)|)$; it is piecewise linear and can have a quadratic number of local minima. Several algorithms that locate a minimizer of f are presented. The best of these has time complexity $O(n^3)$ in the worst case. Our most practical algorithm appears to be one which has worst case behavior of only $O(n^3 \log(n))$, but which we conjecture to have expected time complexity $O((n \log(n))^2)$. Generalizations to k dimensions are mentioned.

1 INTRODUCTION

Given n points (x_i,y_i) in the plane the object is to fit a line which is optimal with respect to the criterion of minimizing the magnitude of the median residual. We are therefore concerned with the minimization of the objective function

(1) $$f(\alpha,\beta) = \text{median}(|y_i - (\alpha + \beta x_i)|).$$

Because order is preserved under monotone transformations, one could just as well have considered the squared residuals, so by analogy with traditional least squares methods, this is called the minimum median squared residual distance funtion. If the global minimum is attained at (α^*,β^*), then $y = \alpha^* + \beta^* x$ is called the MM^{2}R, or minimum median squared residual fit.

One reason for the interest in this fitting procedure is its high breakdown point. Roughly speaking, the breakdown point of a statistical estimator is the smallest percentage of contamination which may cause the estimator to take on arbitrarily large absolute values. This feature of a statistical procedure is emerging as a very important one (see e.g., Donoho and Huber [1], Donoho [2], and the original work on breakdown by Hampel [3]). We do not prove it it here, but the breakdown point of MM^{2}R is 50%, the highest sensible breakdown point that is possible.

The following example illustrates this fact and may help to understand why the MM^{2}R method compares so favorably with least squares with respect to breakdown point. Suppose there are n + 1 points randomly distributed about the line y = 10 and that this sample is contaminated by j points, randomly distributed about the line $y = -t(1 + x)$, $t > 0$.

Even if j is as small as 1, the least squares fit can be altered by an arbitrarily large amount simply by choosing t large enough. Thus least squares will break down by the addition of as few as one contaminating point. However even if j is as large as n, the MM^2R line would not be too badly affected, no matter what t is chosen. It will not break down here, even with contamination approaching 50%.

Previous algorithms for calculating the MM^2R fit were based on gradient and line search methods. The presence of several local minima was recognized and dealt with by repetition of the iterations from several different starting points. Such an algorithm cannot guarantee that the optimal (α^*,β^*) has been found, and it is not even clear how many restarts are appropriate for a desired accuracy of approximation.

In the next section we characterize local minima of f and thus give a necessary condition for global minimizers. This converts the problem of minimizing f into a discrete optimization. In the course of studying the geometry of f, we also show that the continuous optimization approach was, in fact, seriously hampered because f can have $O(n^2)$ local minima.

Two algorithms exploiting our characterization are given. The first of these has time complexity $O(n^3)$ and the second has complexity $O(n^3 \log(n))$. The reason for putting forth the second algorithm is that in the last section it is modified to provide a probabilistic algorithm which may be the most practical one available. On the basis of modest simulation and a reasonably persuasive heuristic argument, it is conjectured to have complexity $O([n \log(n)]^2)$ on the average.

In all the considerations which follow, we suppose that the initial <u>data</u> (x_i,y_i), $1 \le i \le n$, <u>are in general position</u>. More explicitly, we assume no three points are colinear, and no two pairs of points determine parallel lines. We also need to disambiguate the notion of median in even sample sizes. If $u_1 \le \ldots \le u_n$, our convention, based on convenience, will be to take u_m as the median, where $m = 1 + \lfloor n/2 \rfloor$. This reduces to the middle value for n odd and to the <u>high median</u>, or larger of the two middle values, if n is even. One useful consequence of these conventions is that $f > 0$ when n is larger that 4.

2 STRUCTURE AND ALGORITHMS

Given α and β, the line $\ell_{\alpha,\beta} = \{(x,y): y = \alpha + \beta x\}$ defines residuals $r_i(\alpha,\beta) \equiv y_i - (\alpha + \beta x_i)$. When there is no confusion we will simply write r_i. We say that $\ell_{\alpha,\beta}$ "bisects" the three distinct points (x_{i_j},y_{i_j}), $j=1,2,3$, if the residuals r_{i_j} are all the same size and not all have the same sign. If also $x_{i_1} < x_{i_2} < x_{i_3}$ and $r_{i_1} = -r_{i_2} = r_{i_3}$, we say that $\ell_{\alpha,\beta}$ equioscillates with respect to the points. It is easy to see that each triple of points is "bisected" by three lines, one of which equioscillates.

We begin the study of the combinatorics of minimizing f by noting that any line $\ell_{\alpha,\beta}$ partitions $\{1,...,n\}$ into three sets,

$$B_{\alpha,\beta} = \{i: |r_i(\alpha,\beta)| > f(\alpha,\beta)\}$$
$$M_{\alpha,\beta} = \{i: |r_i(\alpha,\beta)| = f(\alpha,\beta)\}$$
$$S_{\alpha,\beta} = \{i: |r_i(\alpha,\beta)| < f(\alpha,\beta)\}$$

of big, median, and small residuals, respectively. When there is no confusion we will drop the subscripts. The following simple result articulates a necessary and sufficient condition for local optimality.

Main Lemma: The pair (α^*,β^*) is a local mimimum of f if and only if the following conditions hold:

(i) $|M_{\alpha^*,\beta^*}| = 3$

(ii) ℓ_{α^*,β^*} equioscillates with respect to the points indexed by M_{α^*,β^*}

(iii) $|B_{\alpha^*,\beta^*}| - |S_{\alpha^*,\beta^*}| \geq 1$

Proof: Fix β and define $r_i(\alpha)$ by $r_i(\alpha) = y_i - (\alpha + \beta x_i)$. Suppose that $|M| = 1$ and that $|r_p(\alpha)|$ is the median-sized residual. If $r_p(\alpha)$ is positive, $|r_p(\alpha)|$ decreases as α increases; and, if it is negative, as α decreases. As α changes, $|r_p(\alpha)|$ remains the median until that point α' at which another residual, r_q, first attains equal size. We may assume $r_p(\alpha')r_q(\alpha') < 0$. Otherwise, we could change α' and decrease both $|r_p(\alpha')|$ and $|r_q(\alpha')|$. Hence, at a local optimum we always have two median sized residuals with opposite signs.

Now we show that (i) is necessary. Suppose (α,β) is a local optimum, and that there are points, say p and q, whose residuals from $\ell_{\alpha,\beta}$ are median sized and of opposite sign. Note that $\ell_{\alpha,\beta}$ contains the point $\mu = [(x_p,y_p) + (x_q,y_q)]/2$. If no other data point has

residual size $|r_p|$, $\ell_{\alpha,\beta}$ may be rotated about $\underline{\mu}$ so that both $|r_p|$ and $|r_q|$ decrease (equally). They will remain median sized residuals up to that point when another residual, say r_s, first attains equal size. This proves that (i) is necessary and also that a line defined by a local optimum will "bisect" the points indexed by M.

In fact if (α,β) is a local optimum, $\ell_{\alpha,\beta}$ must equioscillate. If not, it already passes through the midpoint of two "bisected" points that have opposite sign. By rotating about this midpoint, all three median sized residuals may be reduced in size. Since at least one of these residuals must remain median sized, (ii) is necessary for a local minimum.

To see that (iii) is necessary, suppose that (α,β) is a local minimizer of f. Then (i) and (ii) hold, so there are p,q, and $t \in M$ with $x_p < x_q < x_t$ and $r_p = -r_q = r_t$. The line $\ell_{\alpha,\beta}$ may be rotated about the midpoint of (x_p,y_p) and (x_q,y_q) so that both $|r_p|$ and $|r_q|$ decrease while $|r_t|$ increases. Let (α',β') denote the parameters of the rotated line. By the continuity of $y_i - (\alpha + \beta x_i)$, if (α',β') is close enough to (α,β), $|r_p(\alpha',\beta')|$, $|r_q(\alpha',\beta')|$, and $|r_t(\alpha',\beta')|$ are smaller than all $|r_i(\alpha',\beta')|$, $i \in B_{\alpha,\beta}$ and bigger than all $|r_i(\alpha',\beta')|$, $i \in S_{\alpha,\beta}$. Since $f(\alpha,\beta) < f(\alpha',\beta')$ we must have $M_{\alpha',\beta'} = t$, $S_{\alpha',\beta'} = S_{\alpha,\beta} \cup \{p,q\}$, and $B_{\alpha',\beta'} = B_{\alpha,\beta}$. The sizes of $B_{\alpha',\beta'}$ and $S_{\alpha',\beta'}$ can differ by at most 1, because $|r_t(\alpha',\beta')|$ is a unique median, and these facts now imply (iii).

Finally, (i) – (iii) are clearly sufficient. Otherwise there would exist (α',β') arbitrarily close to (α,β) for which $f(\alpha',\beta') < f(\alpha,\beta)$. But this is impossible because $\ell_{\alpha',\beta'}$ cannot equioscillate.■

Besides characterizing local minimizers of f, the lemma turns the problem into a discrete one. Only lines that equioscillate with respect to some triple of data points need be considered. The following algorithm checks $f(\alpha,\beta)$ on such a line for each triple. By the lemma, the equioscillating line with the smallest value of f will determine (α^*,β^*).

Crude Algorithm:

- $d^* \leftarrow \infty$
- $(\alpha^*,\beta^*) \leftarrow (0,0)$
- for each distinct triple i,j,k
  - renumber points so $x_i < x_j < x_k$
  - $\beta \leftarrow (y_i - y_k)/(x_i - x_k)$; $\alpha \leftarrow [y_j + y_k - \beta(x_j + x_k)]/2$
  - $d_{i,j,k} \leftarrow f(\alpha,\beta)$

- if $d_{i,j,k} < d^*$, $d^* \leftarrow d_{i,j,k}$ and $(\alpha^*,\beta^*) \leftarrow (\alpha,\beta)$
- restore points to original numbering

This may be the first finite, exact algorithm for MM^2R fits. Its crudeness is reflected in its complexity of $O(n^4)$, even if one uses a linear cost method to obtain median($|y_i - (\alpha + \beta x_i)|$) for each of the $n(n-1)(n-2)/6$ lines mentioned. It would not be practical for even as few as 50 points.

This crude algorithm nevertheless contains the germ for a useful method with a faster running time. The first step is to focus on lines through pairs of points. Let $\beta_{i,j} = (y_i - y_j)/(x_i - x_j)$. Because equioscillation is necessary, the optimal β^* must equal $\beta_{i,j}$ for some $i \neq j$. So regard β in (1) as fixed, and consider the reduced problem of finding a minimizer of

$$(2) \qquad g(\alpha) = \text{median}(|z_i - \alpha|)$$

where $z_i = y_i - \beta x_i$. If the z_i are in increasing order then a little thought convinces one that the minimum value of $g(\alpha)$ is equal to the length of the smallest interval that contains half of the z_i. The minimizer α^*, is then the midpoint of this interval. Thus, writing $m(n)$ for the median of $\{1,\ldots,n\}$,

$$(3) \qquad \begin{aligned} &\text{if } z_{p+m(n)} - z_p = \min[(z_{j+m(n)} - z_j),\ j=1,\ldots,m(n)], \text{ then } \min[g(\alpha)] = z_{p+m(n)} - z_p, \\ &\alpha^* = (z_{p+m(n)} + z_p)/2. \end{aligned}$$

This formula can also be derived by noting that the graph of g is continuous and piecewise linear. Each piece has slope ± 1, the slopes change sign at each z_i, $g(z_1) = z_{m(n)} - z_1$, and $g' < 0$ for $z < z_1$.

For any β, let $\ell_{\alpha^*,\beta}$ denote the best possible line with slope β. This suggests the following

Algorithm 1:

- $(\alpha^*,\beta^*) \leftarrow (0,0)$
- $d^* \leftarrow \infty$
- for each distinct pair r,s
  - $\beta \leftarrow \beta_{r,s}$

- $z_i \leftarrow y_i - \beta x_i$, $i=1,\ldots,n$
- sort the z_i
- if $z_{p+m(n)} - z_p = \min(z_{j+m(n)} - z_j)$, $d \leftarrow z_{p+m(n)} - z_p$
- $\alpha \leftarrow (z_{p+m(n)} + z_p)/2$
- if $d < d^*$, $d^* \leftarrow d$ and $(\alpha^*,\beta^*) \leftarrow (\alpha,\beta)$

The complexity of the inner loop is O(n log(n)) due to the sort, and therefore Algorithm 1 has complexity $O(n^3 \log(n))$ overall.

To go beyond Algorithm 1, it is necessary to explicate more of the geometry than was brought out in the Main Lemma. In condition (ii), let us weaken equioscillation to the requirement that $\ell_{\alpha,\beta}$ only "bisect" points indexed by M. If (α,β) also satisfies conditions (i) and (iii), we say it is a "possible local minimum". The next result identifies and counts the "possible local minima".

> **Little Lemma.** There are exactly $n(n-1)/2$ choices of (α,β) such that $\ell_{\alpha,\beta}$ "bisects" the points in $M_{\alpha,\beta}$ and the conditions (i) and (iii) hold.

Consider the line $L_{j,k}$ given by

$$y = \beta_{j,k}(x - x_j) + y_j$$

and which contains points j and k. There are $n - 2$ nonzero residuals from this line. Either at least $m = \lfloor (n-2)/2 \rfloor$ are positive, or m are negative. Assume there are m positive ones (the negative case can be handled similarly). The first step is to find p such that r_p is the m-th smallest positive residual from $L_{j,k}$.

Now consider the line $L^+_{j,k}$ parallel to $L_{j,k}$ and containing (x_p,y_p). There are three points on $L_{j,k}$ and $L^+_{j,k}$, $m - 1$ points **between** the lines, and $q = n - m - 2$ points **not between** them. Consider the line $L^*_{j,k}$ which passes halfway between $L_{j,k}$ and $L^+_{j,k}$. The construction assures that it "bisects" points j,k, and p, that j,k,p are in M, and that (i) and (iii) hold.■

It is natural to ask how many of the "possible local minima" are in fact, local minima. The answer depends on the particular configuration of data points. In fact for the "bisecting" line $L^*_{j,k}$ to equioscillate, all that is required is that x_p lie between x_j and x_k. Using this observation, we exhibit a configuration where f has a quadratic number of local minima.

Let A denote $\lfloor n/4 \rfloor$ points with the smallest x-coordinates and B, $\lfloor n/4 \rfloor$ points with the largest x-coordinates. The remaining points have "middle half" x-coordinates. They will be assigned y-coordinates larger than those of any of the points in A or B. Now, if point $j \in A$ and point $k \in B$ are used to define $L_{j,k}$, as in the proof of the little lemma, the point p will not be in A or B. The line $L^*_{j,k}$ will equioscillate and therefore define a local miminum of f. There are $n^2/16$ such lines, one for each pair j,k in A and B, respectively. This proves the interesting

Corollary. f can have $O(n^2)$ local minima.

The proof of the little lemma suggests another algorithm. Given j and k, compute the line $L^*_{j,k}$ given by

$$y = \beta_{j,k}x + [y_j + y_p - \beta_{j,k}(x_j + x_p)]/2.$$

The number p corresponds to the m-th least positive (or negative) residual from the line through points j and k, and may be obtained in linear time. For some pair j,k, one of these lines will define the minimizer in (1). It may be found in time $O(n^3)$.

Algorithm 2:

- $(\alpha^*,\beta^*) \leftarrow (0,0)$
- $d^* \leftarrow \infty$
- $m \leftarrow \lfloor (n - 2)/2 \rfloor$
- for each pair r,s
  - $d \leftarrow \infty$
  - $\beta \leftarrow \beta_{r,s}$
  - $z_i \leftarrow y_i - y_r - \beta(x_i - x_r)$, $i=1,\ldots,n$
  - $\Pi \leftarrow \{i: z_i > 0\}$, $N \leftarrow \{i: z_i < 0\}$
  - $z_p \leftarrow$ either m-th smallest z_i, $i \in \Pi$ or m-th smallest z_i, $i \in N$
  - $\alpha \leftarrow [y_r + y_p - \beta(x_r + x_p)]/2$
  - if $|z_p| < d^*$, $d^* \leftarrow |z_p|$ and $(\alpha^*,\beta^*) \leftarrow (\alpha,\beta)$

This algorithm checks all "possible local minima". In contrast, Algorithm 1 checks a sequence of lines, each of which is optimal amongst all lines with the same slope. In the next short section we will improve Algorithm 1, both deterministically and on the average.

3 DETERMINISTIC AND RANDOM IMPROVEMENTS

One can improve the practical performance of Algorithm 1 by applying what we call the **wedge trick**. Algorithm 1 loops over all pairs r and s but, as we shall see, the cost of many of these loops can be reduced from nlog(n) to log(n). On the basis of our experience, this reduction is possible often enough to make a big difference in practice.

When examining r,s in the inner loop of Algorithm 1, we find the optimal intercept for lines of slope $\beta_{r,s}$. This defines α^* satisfying

(4) $\quad f(\alpha^*,\beta) \le f(\alpha,\beta)$, all α.

Equation (3) implies that $|M| \ge 2$ and that there are p,q $\in$ M such that $r_p r_q < 0$. If $\ell_{\alpha^*,\beta}$ doesn't satisfy (*), it may be rotated clockwise about $\underline{\mu} = ((x_p,y_p) + (x_q,y_q))/2$ until a point, say s, first satisfies $|r_p| = |r_s|$. Let the slope of this line be β_L. Now rotate $\ell_{\alpha^*,\beta}$ counter-clockwise about $\underline{\mu}$ until a point, say t, first satisfies $|r_p| = |r_t|$ and denote the slope of this line by β_U. This defines a wedge of lines through $\underline{\mu}$ with slopes in the interval $W = [\beta_L,\beta_U]$ for which both p and q remain in M. Therefore for any $\beta \in W$, the optimal line with slope β is in the wedge. This observation allows one to eliminate from further consideration those r,s for which $\beta_{r,s} \in W$.

The modification to Algorithm 1 (call it **Algorithm W**) is to compute $W = [\beta_L,\beta_U]$, take (α,β) to be the parameters of the better of the two lines defining the wedge (e.g., if $|r_s| < |r_t|$, $\beta \leftarrow \beta_L$), and then eliminate all r,s for which $\beta_{r,s} \in W$. The wedge may be computed in linear time simply by solving n-2 simple equations that require $|r_i|$ to be equal to $|r_p|$, i different from p or q. To facilitate the elimination of some $\beta_{r,s}$, a preprocessing step could compute all the $\beta_{r,s}$ and sort then in time $O(n^2 \log(n))$. Then, given W, those $\beta_{r,s} \in W$ may be obtained (say by binary search) in time O(log(n)) and eliminated from further consideration. In the sorted list of $\beta_{r,s}$, we could maintain flags for those slopes which have been eliminated. Given r and s, a search requiring log(n) time reveals whether $\beta_{r,s}$ is flagged. If not, we must process this slope. The optimal intercept for $\beta_{r,s}$ and the wedge generated by $\ell_{\alpha^*,\beta_{r,s}}$ may be obtained in time O(n log(n)). Therefore the total cost of algorithm W is

$$KO(n \log(n)) + (n^2 - K)\log(n) + O(n^2 \log(n)),$$

where K denotes the total number of r,s pairs that needed actual processing.

Monte-Carlo studies of Algorithm W were performed for various models governing random distributions of points and for various values of n ranging from 10 to 125, each combination replicated several times. A rough exploratory analysis of these experiments seems to suggest that O(n log(n)) is a typical value for K. Although there is no thoroughly defensible model for a "random" fitting problem, we make the following concrete statement.

> **Conjecture 1.** If (X_i,Y_i), $1 \le i \le n$, are independent observations from any bivariate distribution with continuous density, then Algorithm W has expected time complexity $O((n \log(n))^2)$.

The probability theory involved in this conjecture appears to be nontrivial. On the practical side, two points are in order:

- Algorithm W has space complexity of $O(n^2)$. This means that examples of size 1000 become unrealistic without use of external storage.
- Our experience with data sets of size ~120 shows that (α^*,β^*) can be computed rapidly. John Tukey suggested that a useful approximation to the MM^2R fit for 1000 points could be defined by averaging the fits obtained on random subsets of size ~100.

The final algorithm, **Algorithm R**, randomizes the wedge trick. In preprocessing, slopes $\beta_{r,s}$ are computed and sorted. Then, instead of checking the $\beta_{r,s}$ in order, one is selected at random. If it has not been eliminated already, the wedge $W = [\beta_L,\beta_U]$ generated by $\beta_{r,s}$ is computed. Then all $\beta \in W$ are eliminated and the process repeated until all $\beta_{r,s}$ have been eliminated or checked.

Algorithm R appears to perform well. We have no proveable complexity results to assert, e.g., that it is superior to Algorithm 1. However we have put together several heuristic arguments that can be construed as support for the following statement.

> **Conjecture 2.** For Algorithm R, the expected value of K is O(n log(n)).

We know much less about the general problem. If the $(\underline{x}_i,y_i)$ are now n points in R^{k+1}, the MM^2R objective function is

$$\text{median}(|y_i - (\alpha_0+\alpha_1 x_{i1}+\ldots+\alpha_k x_{ik})|)$$

Local minimizers must be the normal vectors to hyperplanes that equioscillates with respect to k + 2 points with median sized residuals. An analogue of Algorithm 1 has complexity $O(n^{k+1} \log(n))$.

References

[1] Donoho, David L. and Huber, Peter J.
The Notion of Breakdown Point.
In Lehman, E. (editor), *Recent Advances in Statistics*. , 1983.

[2] Donoho, David L.
Breakdown Properties of Multivariate Location Estimators.

[3] Hampel, Frank.
Contributions to the Theory of Robust Estimation.
PhD thesis, University of California, Berkeley, 1968.