L_1-*Statistical Procedures and Related Topics*
IMS Lecture Notes – Monograph Series (1997) Volume 31

A comparison of two LP solvers and a new IRLS algorithm for L_1 estimation

C.J. Adcock

University of Westminster, U.K.

N. Meade

Management School Imperial College, U.K.

Abstract: This paper is a comparison of two methods for computing L_1 estimates of the parameter vector β in the linear model. The main methods in the comparison are in two groups: special purpose linear programming (LP) methods which exploit the structure of the objective function and iteratively re-weighted least squares (IRLS). The special purpose LP methods included in the review are: (i) the Barrodale and Roberts (BR) algorithm and (ii) the modified form due to Bloomfield and Steiger (BS). The IRLS methods is a new development which exploits the piecewise differentiability of the objective function and which avoids the difficulties previously associated with least squares based schemes. All algorithms have been implemented in a common language, in order to provide a better basis for comparison. To summarise: we found that our implementations of the BR & BS algorithms are generally quicker than existing implementations and general purpose LP solvers; the new IRLS algorithm is faster in circumstances where the number of observations is very large relative to the number of parameters to be estimated.

Key words: Regression, linear model, minimum absolute deviations, LP solvers, iteratively re-weighted least squares, piecewise differentiability.

AMS subject classification: 65J05, 62G05.

1 Introduction

The method of minimum absolute deviation (MAD) or L_1 estimation, to give it one of the many names by which the technique is known, is a robust

method for estimating the parameters in the linear model:

$$y = X\beta + \varepsilon.$$

The objective function to be minimised is:

$$f(\beta) = \sum_{i=1}^{n} |y_i - \beta^T x_i| \tag{1}$$

where $\beta \in \mathcal{R}^k$. Minimising $f(\beta)$ will also give maximum likelihood estimators of β when the $\{y_i\}$ are random sample from the double sided exponential distribution. In this paper it is assumed that the function is not degenerate, in which case it will possess a unique minimiser, β^* say, at which up to k of the residuals:

$$r_i = y_i - \beta^{*T} x_i \tag{2}$$

will satisfy:

$$r_i = 0, \ i \in B, \text{ say} \tag{3}$$

The set B defines a set of basis vectors x_i which span $\mathcal{R}^k$. If the subsets of $\{y_i\}$ and $\{x_i\}$ defined by B are denoted by y^* and X^*, then β^* satisfies:

$$X^*\beta^* = y^* \tag{4}$$

There are two general types of algorithm for calculating MAD estimates of the parameters of a linear model. The first type relies on the fact that the objective function $f(\beta)$ at equation (1) can be formulated as a linear program, Charnes et al (1955). This type of method includes a procedure due originally to Barrodale and Roberts (1973) [henceforth BR] which exploits the fact that the MAD objective function may be written as an LP with special structure. Other LP methods, which also exploit the special structure, have been reported by Bloomfield & Steiger (1980) and Seneta & Steiger (1984). The second type of procedure uses iteratively re-weighted least squares (IRLS). This method was reported by Schlossmacher (1973) and Fair (1974). According to Bloomfield and Steiger (1984, page 259) [henceforth BS], however, it was due originally to Beaton & Tukey (1974), Comparative studies are reported in BS. There is a comprehensive review of algorithms in Dielman (1992).

One of the motivations for this paper is that, although there are very strong similarities between the special purpose LP solvers mentioned above, comparisons are limited by the fact to date, and to the best of our knowledge, the software implementations are quite distinct. Self evidently, comparison is greatly facilitated if the software is written in the same language

and meets similar design criteria. A second motivation is our wish to develop algorithms to minimise mixed objective functions of the form:

$$f(\beta) = \alpha \sum_{i=1}^{n} (y_i - \beta^T x_i)^2 + (1-\alpha) \sum_{j=1}^{m} |Y_j - \beta^T X_j|. \qquad (5)$$

These objective functions arise in other robust methods and in dynamic estimation schemes in which the current parameter estimates are (approximately or asymptotically) normally distributed. Robust estimation methods using similar convex objective functions, in which there is a modulus term, have been studied by Dodge & Jureckova (1991 and 1992) and related computational aspects are reported in Dodge et al (1991). Objective functions of the form in (5) also arise in portfolio optimisation when the conventional quadratic programming formulation is extended by the inclusion of transactions costs. In the case of portfolio optimisation, the minimisation of $f(\beta)$ is invariably carried out subject to a number of linear inequality constraints of the values of the parameter vector β. Algorithms to minimise $f(\beta)$ given at equation (5) may use IRLS methods - see Adcock & Meade (1995) for an example of IRLS used in portfolio optimisation. However, the well reported deficiencies of IRLS have prompted us to consider the general question of algorithms for MAD estimation ab initio.

The purpose of this paper is therefore to compare the solutions times of the main established special purpose LP and IRLS algorithms for MAD estimation. We use new implementations of the Barrodale and Roberts and the Bloomfield and Steiger algorithms. We also present a new procedure for IRLS. The algorithm that we describe in Section 3 of this paper is different from the scheme developed by Schlossmacher and others in that it exploits the piecewise differentiability of the objective function. All algorithms included in the comparison have been re-implemented in a single programming language and in a similar programming style. The aim is to provide a fair basis for comparison of the solution times. In addition, and as reported below, we have found that our new code offers performance improvements over existing software.

The structure of the paper is as follows. Section 2 describes methods based on special purpose linear programming methods. Section 3 presents our scheme which uses iteratively re-weighted least squares. We compare performance using a number of different data sets in Section 4. The final section of the paper contains a summary and concluding remarks.

2 Linear programming methods

Following Charnes et al (1955), the objective function at (1) may be written

exactly as:

$$f(\beta) = \sum_{i=1}^{n} |y_i - \beta^T x_i| = \sum_{i=1}^{n} |e_i^+ + e_i^-| \tag{6}$$

where:

$$e_i^+, e_i^- \geq 0 \tag{7}$$

and where the corresponding n-vectors e^+ and e^- satisfy:

$$y - X\beta = e^+ - e^- \tag{8}$$

Minimisation of $f(\beta)$ is now a linear programming problem involving the $2n+p$ variables e^+, e^- and β, together with the n equality constraints at (8) and the $2n$ non-negativity constraints at (7). Calculation of the minimiser β^* may be undertaken using standard LP solvers. However, these methods are very slow when compared with special purpose solvers which can exploit the structure of the LP formulation of the objective function. The following results are summarised from BS who provide further details and proof of the properties of the algorithms.

2.1 The Barrodale and Roberts (BR) algorithm

The BR algorithm may be viewed as row and column operations on a $(n+k) * (k+1)$ matrix A. The initial value of A is:

$$A = \begin{bmatrix} X & y \\ I & 0 \end{bmatrix}$$

In the steps below the elements of A are $\{a_{ji}\}$. Note that in this notation i indexes columns and j rows rather than the more conventional arrangement.

Step 1 compute:

$$\begin{aligned} g_i &= \sum_j |a_{ji}|, \; j \text{ over } a_{jk+1} = 0 \\ h_i &= \sum_j a_{ji} \cdot sign(a_{jk+1}) \\ l_i &= \min(g_i - h_i, g_i + h_i) \end{aligned}$$

where: $i = 1(1)k$ and $j = 1(1)n$.

Step 2 determine:

$$p = I \text{ where } l_I = \min_i(l_i)$$

If $l_I > 0$ then go to **Step 5**

Step 3 determine the pivot row q by finding the MAD estimate of t in:

$$f = \sum_j |a_{jk+1} - ta_{jp}|$$

ie, find $t^* = y_q/x_{qp}$. It should be noted that this requires a sort routine and that suitable choice of sorter can affect algorithm timings.

Step 4 pivot on row q column p, ie compute new columns $a'_{\cdot j}$ of A:

$$\begin{aligned} a'_{\cdot p} &= a_{\cdot p}/a_{qp} \\ a'_{\cdot j} &= a_{\cdot j} - a_{qj}a_{\cdot p} \; j \neq p \end{aligned}$$

and go to **Step 1.**

Step 5 The minimiser β^* may be recovered from the block of A corresponding to the initial zero vector, ie in column $k+1$ rows $n+1$ through $n+k$, together with a sign reversal. Specifically: $\beta_i^* = -a_{n+i,k+i}$.

2.2 Bloomfield and Steiger (BS) algorithm

There are two modifications to the BR algorithm which are described in BS. One is due to Bloomfield & Steiger (1980) themselves. It is the essentially the same as BR except that in Step 1 a heuristic is used to compute g_i and h_i. In an obvious notation:

$$g_i^{BS} = g_i^{BR}/\sum_j |a_{ji}|, \; h_i^{BS} = h_i^{BR}/\sum_j |a_{ji}|.$$

According to Bloomfield & Steiger, the BS algorithm often converges more quickly than the original BR procedure.

3 Algorithms based on iteratively reweighted least squares (IRLS)

Iteratively re-weighted least squares (IRLS) is an alternative approach to LP. The usual approach is to write $f(\beta)$ identically as:

$$f(\beta) = \sum_{i=1}^{n} (y_i - \beta^T x_i)^2/|y_i - \beta^T x_i| \tag{9}$$

If β_p is an approximation to β^*, a new approximation is computed by minimising the sum of squares:

$$f(\beta) = \sum_{i=1}^{n} (y_i - \beta^T x_i)^2/|y_i - \beta_p^T x_i| \tag{10}$$

which is equivalent to differentiating (9) while holding the $|y_i - \beta^T x_i|$ terms in the denominator fixed. In conventional OLS matrix notation, the equation for the new approximation β_{p+1} may be written as:

$$X^T W_p X \beta_{p+1} = X^T W_p y \tag{11}$$

where:

$$W_p = Diag\{1/|y_i - \beta_p^T x_i|\} \tag{12}$$

Since it is known that, at the minimiser β^*, up to k of the residuals $y_i - \beta^{*T} x_i$ will equal zero, this procedure requires some modification or it will fail as the elements of W_p become very large. A common modification is to define:

$$W_p = Diag\{W_{pi}\}$$

where:

$$\begin{aligned} W_{pi} &= 0 \text{ if } y_i - \beta_p^T x_i = 0 \\ &= 1/\left|y_i - \beta_p^T x_i\right| \text{ otherwise} \end{aligned} \tag{13}$$

This algorithm does not run without problems in practice and it is criticised in BS on the grounds that it is slow and prone to be unstable. A modification to the basic IRLS scheme was introduced by Adcock & Meade (1995). They note that at the points at which it exists, the vector of partial derivatives of $f(\beta)$ is:

$$f'(\beta) = 2X^T W X - 2X^T W y - (\sum_A x_i - \sum_B x_i) \tag{14}$$

where:

$$W = Diag\{1/|y_i - \beta^T x_i|\} = Diag\{W_i\} \tag{15}$$

$A = \{i : (y_i - \beta^T x_i) < 0\}$ and $B = \{i : (y_i - \beta^T x_i) > 0\}$. This suggests the iterative scheme with limiting equation:

$$\beta = (X^T W X)^{-1}\{X^T W y + 0.5(\sum_A x_i - \sum_B x_i)\} \tag{16}$$

which may be re-arranged as:

$$\begin{aligned} \beta_{p+1} &= \beta_p - 0.5(X^T W_p X)^{-1}(\sum_A x_i - \sum_B x_i) \\ &= \beta_p - 0.5\delta_p \text{ say,} \end{aligned} \tag{17}$$

where δ_p is the step length at iteration p. For practical purposes, the modification of W_p described at (13) is employed. The process terminates

when the absolute change in the value of the objective function is less than a given tolerance and the absolute change in each estimated parameter value is also less than a set tolerance.

To improve convergence, we also employ a number of empirical modifications of the scheme. When the process described above converges, to β' say, the label set corresponding to the k smallest absolute values of the residuals $|y_i - \beta'^T x_i|$ is used to define a basis B^*. If the subsets of $\{y_i\}$ and $\{x_i\}$ defined by B^* are collectively denoted by y^* and X^*, then the minimiser β^* is computed as the solution to:

$$X^*\beta^* = y^* \tag{18}$$

as long as $f(\beta^*) < f(\beta')$. Otherwise the minimiser is taken as β'. It should be noted that if $f(\beta^*) \geq f(\beta')$ then β^* cannot be the minimiser which implies that β' is not the minimiser either. However, for the data sets described below, this algorithm always converged to the correct solution. That is, the solution computed by the IRLS method described above was always the same as that computed by the BR or BS algorithms. In this context the same is taken to mean an accuracy equal or better than the process termination parameters.

4 A comparison of performance

In order to compare the computational efficiency of the MAD algorithms described, the times taken by the algorithms to solve a range of problems were measured. If a general data set is denoted as $\{y_i; x_{ij};\ i = 1(1)n,\ j = 1(1)k\}$ then the test data was generated by the following procedure.

1. The x_{i1} are sampled from a uniform distribution on the interval (-1000, 1000), for $i = 1(1)n$.

2. The remaining x_{ij} , $j = 2(1)k$ are generated by the equation:

$$x_{ij} = u_{ij} + c_j x_{1j}$$

where the $u_{ij} \sim U$(-1000, 1000) for $i = 1(1)n$, $j = 2(1)k$ and where the constants c_j were set to 2.0, $j = 2(1)k$, to control the collinearity between the x_{ij}.

3. The dependent variables were generated by the equation:

$$y_i = \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i$$

where the true coefficients $\beta_j \sim U$(-1, 1).

MAD estimation corresponds with the maximum likelihood estimation of the parameters $\{\beta_j\}$ where the residuals ε_i follow a double sided exponential distribution. BS use both the Gaussian and the Pareto in their algorithm comparisons. This led us to generate three different data sets where the values of the error term ε_i are sampled from three different distributions, namely the double sided exponential, the Gaussian and the Pareto with appropriate parametrisation. In order to standardise the error distributions, the interquartile range for each distribution was set to (-50, 50). Data sets were generated with the combination of numbers of observations and numbers of variables shown in Table 1. For each error distribution, four sets of observations for each combination marked with tick, giving 480 = 3 * 4 * 40 data sets in all.

Table 1: Data sets generated

		Number of Observations									
		10	20	50	100	200	500	1000	2000	5000	10000
Number of variables	1	*	*	*	*	*	*	*	*	*	*
	2	*	*	*	*	*	*	*	*	*	*
	5	*	*	*	*	*	*	*	*		
	10		*	*	*	*	*	*	*		
	20			*	*	*	*	*			

The BR algorithm and the iteratively re-weighted least squares (IRLS) algorithm were used to estimate the parameters of each data set. Both algorithms were programmed in Fortran 77. We implemented a new version of the BR algorithm. This follows the procedure described in Section 2. The computations were performed on a Silicon Graphics workstation. Since the solution times for small problems was very short, the actual time measured was that to solve the same problem ten times.

The convergence parameters in the IRLS algorithm were set so that the algorithm was deemed to have converged if:

$$f(\beta_{p-1})/f(\beta_p) - 1 \leq 10^{-6} \text{ and } \frac{1}{k}\sum_{j=1}^{k} |\beta_{j,p} - \beta_{j,p-1}| \leq 10^{-6} .$$

It is well known that the values of these tolerances can affect solution time substantially. The above values were chosen after some initial investigation with the aim of ensuring that, for those data sets where the new IRLS method converged satisfactorily, the numerical discrepancies between IRLS and LP solutions were small. As already reported in Section 3, for the data sets considered, all three method always converged to the same solution.

In order to gain some understanding from these timings, a model of the computation time, T say, as a function of the number of variables, k say, and the number of observations, n say, was constructed. Anderson & Steiger (1982) proposed a model of the form:

$$T = \gamma_0 + \gamma_1 n + \gamma_2 k + \gamma_3 nk + \eta$$

Trials with this model were unsatisfactory in that the estimated values of T were negative for small values of k. Beasley (1990) suggested a log-linear model for the timing of the LP solutions. This provided a basis for the following model:

$$\ln(T) = \gamma_0 + \gamma_1 \ln(k) + \gamma_2 \ln(n) + \gamma_3 \ln(\ln(n)) + \eta'$$

The coefficients η were estimated by minimising $\sum |\eta'|$. The estimated coefficients for the BR algorithm and for IRLS are in Table 2 which shows the estimated coefficients over all data sets and for each error distribution separately.

Table 2: Estimates of parameters in timing model

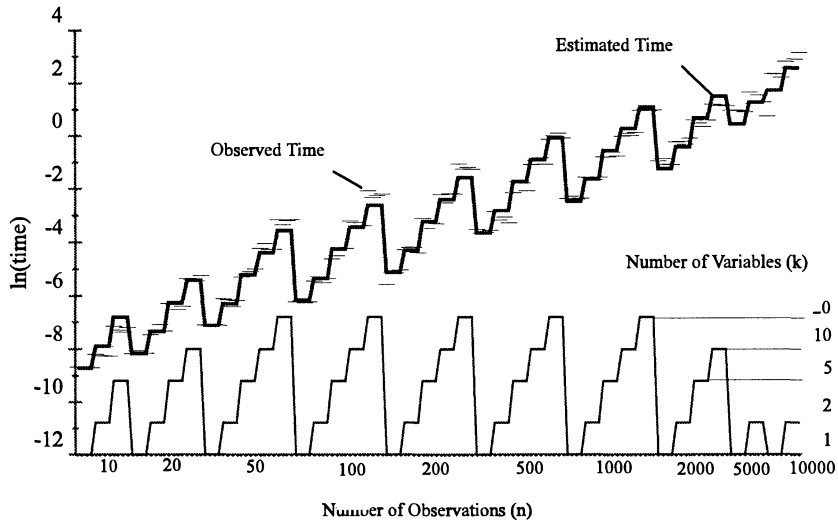
Error Distribution		Barrodale and Roberts	Iterated Least Squares
Neg. Exp.	0	-10.76	-9.57
	1	1.19	1.58
	2	2.32	1.29
	3	-3.96	-0.89
	mean $\lvert e' \rvert$	0.20	0.38
Gaussian	0	-10.68	-9.51
	1	1.18	1.58
	2	2.38	1.17
	3	-4.21	-0.55
	mean $\lvert e' \rvert$	0.19	0.35
Pareto	0	-10.73	-8.59
	1	1.19	1.74
	2	2.35	1.54
	3	-4.15	-2.15
	mean $\lvert e' \rvert$	0.19	0.31
Combined	0	-10.70	-9.30
	1	1.19	1.56
	2	2.34	1.30
	3	-4.08	-1.02
	mean $\lvert e' \rvert$	0.20	0.42

Comparison of the parameters for the time taken by the BR algorithm shows a consistency between the different error distributions. There is no discernible difference in timings caused by the choice of error distribution. The timings for IRLs are more dispersed about the fitted equation, showing that for the same dimension of problem there is twice as much variability in solution time in terms of mean absolute error than for the BR algorithm. For a problem of the same dimensions, the Pareto distribution appears to require a greater solution time than the other distributions by a factor of about two. Anderson & Steiger (1982) conjectured that the timing of the BR algorithm increased with approximately n^2. Their conjecture is confirmed by these results which suggest that timings are proportional to $n^{2.34}$. The term in $\ln(n)$ adjusts this slightly over a wide range of numbers of observations. The timing increases slightly faster than linearly in terms of the number of variables: it is proportional to $k^{1.2}$.

The reliance of the IRLS technique on matrix inversion means that the computation time increases faster than linearly with k. In fact it appears that computation time is proportional to to $k^{1.5}$. However, in contrast to the BR algorithm, the IRLS computation time increases only a little faster than linearly in the number of observations, ie it is proportional to $n^{1.3}$.

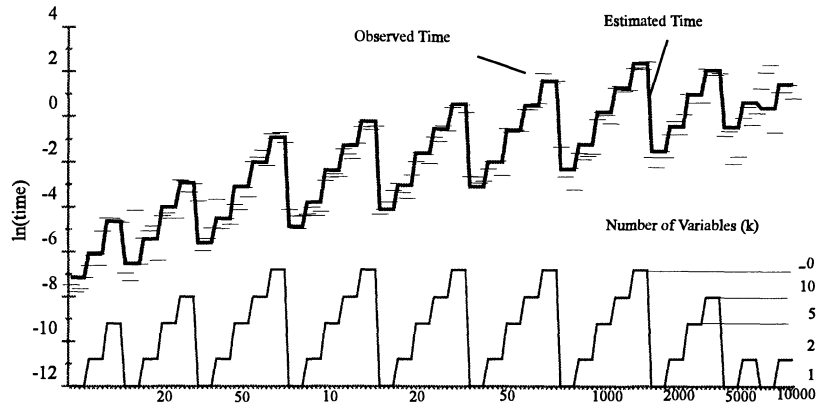
The actual times plotted along with the estimated times for the BR and IRLS algorithms are shown in Figures 1, 2 and 3.

Figure 1. Times for Barrodale Roberts



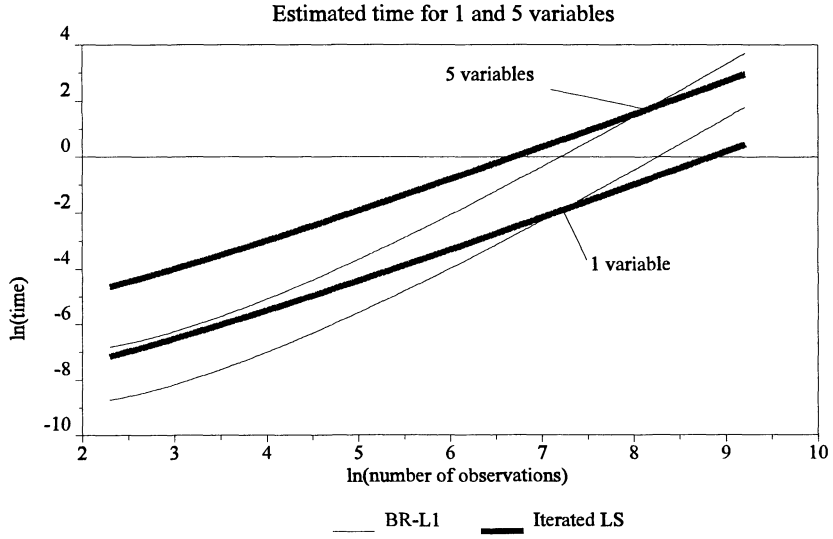
Times are shown for n observations and k variables, where n and k are identified respectively by the horizontal axis and the thin line above

Figure 2. Times for IRLS



Times are shown for n observations and k variables, where n and k are identified respectively by the horizontal axis and the thin line above

Figure 3
Estimated time for 1 and 5 variables



The greater variability of the IRLS solution times is apparent. More interestingly, the superiority of the BR algorithm decreases as the number of observations increases. This crossover is clearly brought out in Figure 4 where the expected solution times for each algorithm is shown for one and five variables. Thus, for one variable and about 1500 observations, the expected solution times are the same. For a larger number of observations,

one would expect IRLS to be quicker. For five variables, the cross over occurs at about 3300 observations. We note the results of a simulation study of times for MAD estimation of simple linear regression reported by Armstrong & Frome (1976), who also compared IRLS and LP methods. Although their results cannot be compared directly with ours, they also found a similar superiority of LP over IRLS as far as mean solution time is concerned. However, as the table in their paper indicates, the superiority declines as the number of observations increases.

As indicated above, as part of this study we wrote up a fresh modular implementation of the BR algorithm. This has a built in facility to be converted to the Bloomfield and Steiger algorithm which we also used to compute solution times for the above data sets. The parameter estimates in the above timing model for the BS algorithm are shown in Table 3.

As the table shows, there is no discernible difference between the two solution times for the BR and BS algorithms. The estimated value of γ_2 for the BS algorithm is broadly consistent with Bloomfield and Steiger's contention that the computation time is linear in the number of observations n. However, the more interesting coefficient is that for the number of variables which shows that the solution time increases less than linearly. It is in fact proportional to $k^{0.67}$. The disadvantage with this implementation of the BS algorithm is that it is slow for small problems.

Table 3: Estimates of parameters in timing model for BR/S algorithm

Error Distribution		BR/S
Combined	0	-2.07
	1	0.67
	2	1.21
	3	-4.11
	mean $\lvert e' \rvert$	0.44

Finally, we also used two other MAD algorithms which are in the public domain and computed the solution times. The first was the version of the BR algorithm provided in the NAG library. The second was a line by line re-implementation of the BR algorithm as described in Barrodale and Roberts (1974). We found that, with respect to the number of variables, these two algorithms are similar to our modular implementation of BR and BS in that the solution times vary linearly with the number of variables. However, we also found that the solution time of these public domain algorithms increases much more steeply with the number of observations.

It should be noted that none of the above data sets are degenerate and that, because of the use of the uniform distribution, there are no extreme outliers in the independent x variables. Investigation of conditions in which

the new IRLS algorithm, or indeed BR/BS, might fail remains a topic for further investigation. At present, we are inclined to the view that some algorithm problems may be due in part to the use of an inappropriate programming language.

5 Summary and concluding remarks

In this paper, we have presented and compared two methods for MAD estimation of the parameters in a linear model. They are described in detail, with accompanying pseudo-code in Adcock & Meade (1997). There is no single algorithm that is superior to all the others, at least as far as the data sets that we have investigated are concerned, The IRLS algorithm that we have described in this paper did not suffer from any problems of convergence or of numerical accuracy. Furthermore, we found it to be faster than all of the special purpose LP algorithms for data sets where the number system is greatly over-determined. We did not find significant differences between the BR and BS algorithms. Our investigations of various implementations of the BR algorithm indicated that the speed of the sort routine, which is a necessary step in each iteration of the algorithm, is crucial to the overall time taken. According to Dielman (1992), it is also likely that the algorithm due to Armstrong et al (1979), which employs an LU decomposition of the basis matrix, may lead to further performance improvements. This remains a topic for further investigation.

References

[1] Adcock C. J. and Meade N. (1995). A simple algorithm to incorporate transactions costs in quadratic optimisation. *European Journal of Operational Research* **79**, 85-94..

[2] Adcock C. J. and Meade N. (1997). Two algorithms for minimum absolute deviation estimation in the linear model. In preparation.

[3] Anderson D. and Steiger W. L. (1982). A comparison of methods for discrete L1 curve fitting. Technical Report 69, Department of Computer Science, Rutgers University.

[4] Armstrong R. D. and Frome E. L. (1976). A comparison of two algorithms for absolute deviations curve fitting. *J Am. Statist. Assoc.* **71,** 328-330.

[5] Armstrong R. D., Frome E. L. and Kung D. S. (1979). A revised simplex algorithm for the absolute deviation curve-fitting problem. *Commun. Statist. - Simul. Comp.* B **8**, 175-190.

[6] Barrodale I. and Roberts F. D. K. (1973). An improved algorithm for

discrete L1 linear approximation. *SIAM J. Numer. Anal.* **10,** 839-848.

[7] Barrodale I. and Roberts F. D. K.(1974). Algorithm 478. Solution of an overdetermined system of equations in the L1 norm. *Commun. ACM* **14,** 319-320.

[8] Beasley (1990). Linear programming on Cray supercomputers. *J. Opl. Res. Soc.* **41,** 133-139.

[9] Beaton A. E. and Tukey J. W. (1974). The fitting of power series, meaning polynomials, illustrated on band spectographic data. *Technometrics* **16,** 147-185.

[10] Bloomfield P. and Steiger W. L. (1980). Least absolute deviations curve fitting. *SIAM J. Scient. Statist. Comput.* **1,** 290-301.

[11] Bloomfield P. and Steiger W. L. (1984). *Least Absolute Deviations: Theory, Applications and Algorithms.* Boston: Birkhauser.

[12] Charnes A. Cooper W. W.and Ferguson R. O. (1955). Optimal estimation of executive compensation by linear programming. *Management Science* **1,** 138-151.

[13] Dielman T. E. (1992). Computational algorithms for least absolute value regression. In *L1 Statistical Analysis and Related Methods,* Ed. Y. Dodge, pp. 311-326. Elsevier Science Publishers.

[14] Dodge Y., Antoch J. and Jureckova J. (1991). Adaptive combinations of least squares and least absolute deviations estimators: computational aspects. *Comput. Statist Data Anal.* **12,** 87-99.

[15] Dodge Y. and Jureckova J. (1991). Flexible L-estimation in the linear Model. *Comput. Statist Data Anal.* **12,** 211-220.

[16] Dodge Y. and Jureckova J. (1992). A Class of estimators based on adaptive convex combinations of two estimation procedures. In *L1 Statistical Analysis and Related Methods,* Ed. Y. Dodge, pp. 31-45. Elsevier Science Publishers.

[17] Fair R. C. (1974). On the robust estimation of econometric models. *Ann. Econ. Soc. Measurement* **3,** 667-677.

[18] Schlossmacher E. J. (1973). An iterative technique for absolute deviations curve fitting. *J Am. Statist. Soc* **68,** 857-865.

[19] Seneta E. and Steiger W. L. (1984). A new LAD curve fitting algorithm: slightly over-determined equation systems in L1. *Discrete Applied Mathematics* **7,** 79-91.