FINITE ALGORITHMS FOR
LINEAR OPTIMISATION PROBLEMS

by

David I. Clark

A thesis submitted to the
Australian National University
for the degree of Doctor of Philosophy

January, 1981

ACKNOWLEDGEMENTS

The work for this thesis was carried out at the Computing Research Group and the Department of Statistics, I.A.S., Australian National University. I gratefully acknowledge the financial assistance of the Australian National University during this time.

I am indebted to Mike Osborne for his help and supervision of my work and for his constructive comments on the draft of this thesis. I would also like to thank Robin Stanton for helpful conversations and David Anderson in providing test programs for the L_1 algorithm, and members of the Computing Research Group and the Department of Statistics for a congenial and constructive work environment.

Thanks are also due to the typist, Mrs. Susan Watson, for care and tolerance at a difficult time, and to my mother, Cherrie, and my wife, Mel, for conscientious proof-reading.

Finally, to Mel, for encouragement and support, thank-you.

PREFACE

Some of the work of this thesis was carried out in collaboration with Dr. Michael Osborne. In particular, Chapters 3 and 5 contain results which were established jointly. Also, Chapters 2 and 3 have been published as Clark (1980) and Clark and Osborne (1980) respectively.

Unless otherwise stated in the text, the work described is my own.

David G. Clark

ABSTRACT

In this thesis we investigate certain linear optimisation problems,

$$\text{minimise} \quad f(\underset{\sim}{x}) \quad \text{subject to} \quad g_i(\underset{\sim}{x}) \geq 0 \,, \qquad i = 1,\ldots,n$$

where the Kuhn-Tucker conditions

(i) $g_i(\underset{\sim}{x}) \geq 0 \qquad i = 1,\ldots,n$

(ii) for some $\underset{\sim}{u} \geq \underset{\sim}{0}$, $\quad \nabla f(\underset{\sim}{x}) = \Sigma u_i \nabla g_i(\underset{\sim}{x})$

(iii) $\underset{\sim}{u}^T \underset{\sim}{g}(\underset{\sim}{x}) = 0$

comprise a set of simultaneous linear equations.

Chapter 1 introduces the problems, the restricted least squares (RLS), M-estimator, and least absolute deviations (LAD) problems, and places them in their context.

In Chapter 2, the RLS problem is examined, and pruning rules developed which transform a rather inefficient branch and bound algorithm into an essentially iterative one. The implementation of the resulting algorithm is considered in Chapter 3 and, by working with dual variables and using orthogonal transformations, the algorithm in its final form is at least competitive with existing algorithms for this problem. An error analysis is also given, showing that the use of dual variables has led to superior numerical properties.

Chapter 4 considers the structure of the M-estimator function. Several speculations are raised and these are answered either negatively by means of a counter example, or positively by

proving a theorem. The broad areas covered by these speculations include the question of non-uniqueness, the connection between the M-estimator and the LAD estimator, what might be called the "proper behaviour" of the function and the function value itself.

Chapter 5 deals with algorithms for calculating the M-estimator. Existing algorithms are surveyed, and two new ones developed. One of them, a continuation algorithm, is examined in detail and numerical results presented. Finiteness is proved for them both.

Finally, in Chapter 6, the LAD problem is considered. The existing algorithms are reviewed and a new one presented which, although proven finite, did not perform competitively on a particular class of example. The thesis concludes with a discussion of why the algorithm failed, how it differs from algorithms which succeeded for that type of example, and how the algorithm may be improved.

TABLE OF CONTENTS

Page

CHAPTER 1

THE PROBLEMS INTRODUCED

For the general mathematical programming problem

(1.1) minimise $f(\underset{\sim}{x})$ subject to $g_i(\underset{\sim}{x}) \geq 0$, $i = 1,\ldots,n$,

the well-known Kuhn-Tucker conditions for a stationary point at $\underset{\sim}{x}^*$ are

(1.2) $g_i(\underset{\sim}{x}^*) \geq 0 \quad i = 1,\ldots,n$

(1.3) for some $\underset{\sim}{u} \geq \underset{\sim}{0}$, $\nabla f(\underset{\sim}{x}^*) = \Sigma u_i \nabla g_i(\underset{\sim}{x}^*)$

(1.4) $\underset{\sim}{u}^T \underset{\sim}{g}(\underset{\sim}{x}^*) = 0$.

If (1.2) to (1.4), which characterise a local optimum, comprise a set of linear equations, the problem is a linear optimisation problem (LOP). LOPs have interest and importance in their own right, in forms as diverse as standard linear programming, quadratic programming, and matching problems. Provided (1.2) to (1.4) provide a bounded set, there is in principle a finite algorithm to find a feasible pair $(\underset{\sim}{x},\underset{\sim}{u})$ satisfying them, although the number of points visited may be very large. However, finiteness and, in general,tractibility are amongst the virtues of LOPs and they are often used as steps within other algorithms to solve more difficult problems. Thus linearly constrained non-linear optimisation problems can be solved by feasible conjugate direction methods which solve an LOP at each step (eg, Best, 1975) and even when the constraints are non-linear, cutting plane methods still solve an LOP at each step (eg, Luenberger, 1973).

It is worth noting that although general methods may exist for a class of problems, it is often the case that a particular member of the class is so structured that a close examination of the problem yields a much simpler method of solution. Well-known examples of this are geometric programming, (Duffin, Peterson and Zener, 1967), where a particularly vile-looking non-linear optimisation problem can be transformed into an LOP, and the transportation problem, (eg, Hadley, 1962) where an LOP can be solved simply without recourse to more general techniques, such as the simplex method.

The problems studied in this thesis, the restricted least squares (RLS), M-estimator, and least absolute deviation (LAD) problems all display this feature, where a close examination of the problem leads to a method of solution which is able to take advantage of the structure of the problem so that an algorithm can be tailored to fit it expressly. This is clearly illustrated by the RLS problem, (Chapter 2) where, starting from a general branch and bound search algorithm, analysis of the problem leads to a vastly reduced search-tree, and then further analysis leads to a finite iterative algorithm far superior to the original search algorithm. Further analysis still (Chapter 3) then indicates an efficient method of implementation of the algorithm which test results have shown to be more than competitive with standard linear programming techniques, and which has the added advantage of superior numerical properties.

The RLS problem is important both in its own right, and because it forms a central step in many algorithms for more general

LOPs, and as such has received a great deal of attention. Typical of the linear-programming based algorithms written specifically for the RLS problem is that of Lawson and Hanson (1974). The overall scheme of their algorithm is to start from an initial primal feasible solution $(\underset{\sim}{x} = \underset{\sim}{0})$, and at each step maintain both primal feasibility and complimentary slackness, terminating when dual feasibility is also achieved. The other approach which has been tried on this problem is to place it in a branch and bound framework. Armstrong and Frome's algorithm (1976) using this technique is not competitive for large problems, and does not appear promising, yet successive refinements of it resulted in a competitive stable algorithm, similar in some ways to the linear-programming based algorithms, but differing from them in that only complimentary slackness is preserved at each iteration, the direction of the algorithm being towards finding a dual feasible point and then testing it for primal feasibility.

The M-estimator is one of a number of statistical measures which have been suggested in an effort to minimise the effect of and identify outlying observations. Although interest in rejection criteria stems back at least one hundred years (eg, Peirce, 1852), the current surge of interest in the so-called robust estimators was catalysed by Tukey in 1960 when he showed that the widely used least squares estimator was in some ways inferior to the least absolute deviation estimator. Strictly speaking, if $\underset{\sim}{x}^*$ minimises $\Sigma\rho(A\underset{\sim}{x}_i - b_i)$ for some function ρ, then $\underset{\sim}{x}^*$ is a maximum-likelihood or M-estimator for the linear model $\underset{\sim}{b} = A\underset{\sim}{x} - \underset{\sim}{\varepsilon}$ under some appropriate assumption of distribution of $\underset{\sim}{\varepsilon}$. The most commonly used function ρ , and the one under study in this thesis is Huber's (1972) function

$$(1.5) \qquad \rho(t) = \tfrac{1}{2}t^2 \qquad |t| \le c$$
$$= c|t| - \tfrac{1}{2}c^2 \qquad |t| > c\ .$$

One of the thrusts of this thesis has been to examine the problems carefully in an effort to understand their underlying structure, and probably the major contribution has been the examination of the M-estimator when detailed properties of it are given for the first time. In particular, the relationship between the M-estimator and the LAD estimator is explored, and the question of uniqueness thoroughly examined. Although several of the theorems developed do not have a direct bearing on algorithm development, two algorithms arise fairly naturally from the study and an understanding of the structure has facilitated proving finiteness for them.

Another area which has experienced a resurgence of interest due to the interest in robust estimation has been least absolute deviation regression. Used in line-fitting models, it predates the least squares method, being used by Boscovitch in 1757, but it was not until 1973 that an efficient algorithm was written by Barrodale and Roberts. Since then several efficient algorithms based on either the simplex method or a gradient method have been developed. A feature of all these algorithms is that they have a full basis at each step, and have been shown by Osborne, 1980, to be in a sense identical. The algorithm presented in this thesis does not necessarily work with a full basis. It is not yet clear how the increased freedom in choice of descent direction should be used, and in its current form the algorithm is not always competitive, but even the failure of the algorithm has helped in understanding the structure of the problem.

A final point is that although the nature of this approach, of studying the structure of the problem, of necessity leads to an algorithm specifically tailored to a particular problem, the approach of an algorithm is not necessarily confined to its own problem. Thus the first algorithm for the M-estimator is the progenitor of that for the LAD estimator, and the approach developed there can be fairly directly applied to solving other LOP problems, and the RLS algorithm bears a close relationship to the second M-estimator algorithm.

CHAPTER 2

AN ALGORITHM FOR THE RESTRICTED LEAST SQUARES PROBLEM

2.1 INTRODUCTION

2.1.1 The Constrained Least Squares Problem

With its wide applicability, the constrained least squares problem (CLSP)

$$(2.1)\qquad \text{minimise}\quad \|A\underset{\sim}{x}-\underset{\sim}{b}\|$$
$$\text{subject to}\quad E\underset{\sim}{x} = \underset{\sim}{f}$$
$$G\underset{\sim}{x} \geq \underset{\sim}{h}$$

has received a great deal of attention. In its equivalent form,

$$(2.2)\qquad \text{minimise}\quad \underset{\sim}{x}^T C\underset{\sim}{x} + \underset{\sim}{c}^T\underset{\sim}{x} + \underset{\sim}{d}^T\underset{\sim}{d}$$
$$\text{subject to}\quad E\underset{\sim}{x} = \underset{\sim}{f}$$
$$G\underset{\sim}{x} \geq \underset{\sim}{b}$$

it is a convex quadratic programming problem, and early algorithms to solve the problem used quadratic programming techniques based on the simplex algorithm (see, eg, Cottle 1968, Cottle and Danzig 1968, Lemke 1968, Wolfe 1959). However, these methods, based on pivoting and inverse basis techniques, have been found to be numerically unstable (see, eg, Wilkinson 1961, 1965, Golub and Wilkinson 1966). Moreover, as shown by Golub 1965, and Golub and Saunders 1969, the problem in its second form (2.2) is always more ill-conditioned than in its first form (1.2).

For these reasons, a number of algorithms have been developed using orthogonalisation procedures. Stoer 1971, uses an L-R decomposition, as do Bartels, Golub and Saunders 1970, whilst Lawson and Hanson 1974, (cited by Bartels, 1975, as the definitive handbook on Least Squares problems) use a Q-R decomposition. The numerical properties of the two decompositions are similar. It should be emphasised that the chief aim of these methods is to improve the numerical stability of the algorithm, and that any improved efficiency (as reported, eg, by Osborne, 1976) is a pleasing side-benefit.

It is instructive to examine the algorithms in an effort to obtain an overview of what is happening within them, and a useful way of doing so is via the Kuhn Tucker (K.T.) conditions, which all algorithms seek to fulfil at the optimum. For the general mathematical programming problem (MPP)

(2.3) minimise $f(\underset{\sim}{x})$ subject to $g_i(\underset{\sim}{x}) \geq 0$, $i = 1,\ldots,m$

the K.T. necessary conditions for $\underset{\sim}{x}^*$ to minimise $f(\underset{\sim}{x}^*)$ are

(2.4) $g_i(\underset{\sim}{x}^*) \geq 0$, $i = 1,\ldots,m$

(2.5) $\exists\ \underset{\sim}{u}^* \geq \underset{\sim}{0}$ such that $\nabla f(\underset{\sim}{x}^*) = \Sigma u_i \nabla g_i(\underset{\sim}{x}^*)$

(2.6) $\underset{\sim}{u}^{*T} g(\underset{\sim}{x}^*) = 0$

These conditions which, in the case of a convex objective function with consistent linear constraints are sufficient for global minimisation, can be described respectively as primal feasibility, dual feasibility and complimentary slackness. Now the simplex method, at each iteration, produces a point which satisfies both primal feasibility (or dual feasibility in the case of the dual simplex method) and complimentary slackness, and proceeds until it also achieves dual feasibility. This feature is present in those quadratic programming algorithms based on the simplex method. It is also present, as far as the author can determine, in all of the algorithms based on orthogonalisation techniques. Complimentary slackness is ensured by optimising a subproblem at each iteration, and either primal or dual feasibility is achieved by careful choice of the subproblem solved, often with a certain amount of programming difficulty, if not computational effort. It is in departing from this requirement that the algorithm below is interestingly, if not significantly, different.

2.1.2 The Restricted Least Squares Problem

The restricted least squares problem (RLSP), also referred to as the non-negative least squares problem

$$\text{(2.7)} \quad \text{minimise} \quad \|\underset{\sim}{A}\underset{\sim}{x}-\underset{\sim}{b}\| \qquad \text{subject to} \quad \underset{\sim}{x} \geq \underset{\sim}{0}$$

is a rather simple case of the CLSP (2.1). It does have applicability in its own right, when the model being examined will not permit non-negative parameters, but its main importance lies in its being used as a subproblem at an iteration in the solution of more general problems. Thus, for example, Bartels 1975, and Haskell and Hanson, 1978, solve an RLSP at each iteration of their CLSP algorithms.

RLS problems can be solved using any of the CLSP algorithms, but their importance and the simple nature of the constraints have led to a number of algorithms specifically written for this problem. There is reference to an algorithm due to Bard by Bartels, Golub and Saunders 1970, but details are sketchy and there is no guarantee of finite termination. Lawson and Hanson 1974, give an algorithm in which complimentary slackness and primal feasibility are maintained, with each iteration differing from the previous one by one constraint changing status. Bartels 1975, uses a similar overall scheme, except that he permits several constraints to change status at each iteration. (His method is designed specifically for large sparse matrices.)

An entirely different approach is advocated by Armstrong and Frome 1976, based on an observation by Waterman 1974. They place the problem in a branch and bound framework, and give an improved pruning rule. Due to the tendency of branch and bound solutions to increase exponentially with problem size, this approach does not appear promising, and indeed experimental results of the Armstrong and Frome algorithm bear out this fear (Table 2.4). However, starting from this point, successive refinements eventually lead to an algorithm which is competitive with the algorithms cited above. The final implementation of the algorithm (Chapter 3) is not dissimilar to the Bard-type algorithms of Lawson and Hanson, and Bartels, but does have the basic difference in that at any iteration of the algorithm, neither primal nor dual feasibility is guaranteed. This is illustrated in the sample problem given later in this chapter.

The remainder of this chapter follows the development of the algorithm, starting from the branch and bound approach. The rest of Section 1 defines notation and introduces the branch and bound method. In Section 2, the Armstrong and Frome algorithm is presented and an improved pruning rule is given. Then the K.T. conditions for optimality are established. A rule is given to find a better feasible solution should a feasible solution be found to be sub-optimal. The complexity of the algorithm is considered in Section 3, and under certain circumstances (always satisfied experimentally), linearity of subproblems solved against problem dimension is proved. The experimental results are presented in Section 4. In Section 5, the extension of the algorithm to similar problems is discussed, including the modifications necessary if the K.T. conditions are not readily available.

2.1.3 Notation

The following notation will be used in this chapter.

m, n represent the dimensions of the data matrix A (m variables, n observations)

J^i represents an index set $J^i \subseteq N = \{1,2,\ldots,n\}$

P^i represents the problem

$$\text{minimise} \quad L(\underset{\sim}{x}) = \| A\underset{\sim}{x} - \underset{\sim}{b} \|$$

$$\text{subject to} \quad x_j = 0 \, , \quad j \in J^i$$

$\underset{\sim}{x}^i$ represents the optimal solution to P^i

(Note that the index set J^i defines P^i and hence $\underset{\sim}{x}^i$)

$\underset{\sim}{y}^i$ represents a feasible solution to both P and P^i, that is

$\underset{\sim}{y}^i \geq \underset{\sim}{0}$ and $y^i_j = 0 \, , \, j \in J^i$

(Note that $\underset{\sim}{y}^i$ will not necessarily be optimal for P or for P^i)

When the term "feasible" is used, it will refer to primal feasibility for P, that is, $\underset{\sim}{x}$ is feasible if $\underset{\sim}{x} \geq \underset{\sim}{0}$

2.1.4 Branch and Bound

The branch and bound method builds up a search tree (each node being a problem, P^i) by increasing the number of variables set to zero as a branch is descended. Thus, if P^j is a descendant of P^i, $J^j \supset J^i$. The root of the tree is the problem P^1 where $J^1 = \emptyset$. An example of a search tree is given in Fig. 2.1.

The main considerations of a branch and bound algorithm are:

(i) Choosing which node to branch on next,

(ii) choosing which descendant of this node to consider (solve) next, and

(iii) making use of any special properties of the problem to detect early fathoming of a branch, that is, recognizing when no descendants of a node will yield a better solution.

All of the above considerations are dealt with in the new algorithm.

2.2 THE NEW ALGORITHM

2.2.1 The Armstrong and Frome Algorithm

Armstrong and Frome's node choice is to branch on the node most recently solved until a feasible $\underset{\sim}{x}^i$ is found, and thereafter to branch on the node, P^j, with the smallest $L(\underset{\sim}{x}^i)$. Their choice of the next variable to be set to zero is the most negative free variable if one exists, otherwise the free variable with the largest numerical value. Their pruning rule states that if a node differs from its parent node in that a variable which was negative in the parent node's optimal solution has been set to zero (for example, nodes 2, 8, 18, and 26, 30, 32 in Fig. 2.1), and either the optimal solution of the node is feasible (for example, nodes 4, 6, 10) or has an $L(\underset{\sim}{x}^i)$ greater than or equal to the best existing feasible solution (nodes 6, 7), then no further branches from the present node need to be considered. In the example of Fig. 2.1 (data in Table 2.1, results in Table 2.2), 32 of the possible 64 nodes were solved.

2.2.2 Improved Pruning Rule

The first improvement to the above algorithm is the new fathoming criterion "at node P^i, it is only necessary to branch on variables x_j for which $x_j^i < 0$".

Lemma 2.1

Given $\underset{\sim}{y}^i \geq \underset{\sim}{0}$. Let $J^i = \{j | y_j^i = 0\}$ define P^i and hence $\underset{\sim}{x}^i$. Then there exists some descendant, P^r, of P^i (i.e. $J^r \supseteq J^i$) for which $\underset{\sim}{x}^r \geq \underset{\sim}{0}$ and $L(\underset{\sim}{x}^r) \leq L(\underset{\sim}{y}^i)$.

Proof

If $\underset{\sim}{x}^i \geq \underset{\sim}{0}$, $\underset{\sim}{x}^r = \underset{\sim}{x}^i$; otherwise let k be such that

$$\frac{y_k^i}{|x_k^i|} = \min\left\{ \frac{y_j^i}{|x_j^i|} \;\middle|\; x_j^i < 0 \right\}$$

Let $J^{i+1} = J^i \cup \{k\}$ define P^{i+1} and $\underset{\sim}{x}^{i+1}$. Let

$$\underset{\sim}{y}^{i+1} = \frac{y_k^i}{y_k^i + |x_k^i|} \underset{\sim}{x}^i + \frac{|x_k^i|}{y_k^i + |x_k^i|} \underset{\sim}{y}^i \geq \underset{\sim}{0} .$$

Then, from the convexity of L and the optimality of $\underset{\sim}{x}^i$ and $\underset{\sim}{x}^{i+1}$,

$$L(\underset{\sim}{x}^{i+1}) \leq L(\underset{\sim}{y}^{i+1}) \leq L(\underset{\sim}{y}^i) .$$

If $\underset{\sim}{x}^{i+1} \geq \underset{\sim}{0}$, $\underset{\sim}{x}^r = \underset{\sim}{x}^{i+1}$. Otherwise, the process is repeated and, at each step,

$$L(\underset{\sim}{x}^{i+j}) \leq L(\underset{\sim}{y}^{i+j}) \leq L(\underset{\sim}{y}^{i+j-1}) \leq \ldots \leq L(\underset{\sim}{y}^i) ,$$

until eventually $\underset{\sim}{x} \geq \underset{\sim}{0}$.

The improved pruning rule now follows.

Theorem 2.1

At any node P^i, it is only necessary to branch on variables x_j for which $x_j^i < 0$ in order to find $\underset{\sim}{x}^r$, a best feasible solution descendant from $\underset{\sim}{x}^i$.

Proof

Let $J_-^i = \{j \mid x_j^i < 0\}$.

Let $\underset{\sim}{x}^r$ be the best feasible descendant of $\underset{\sim}{x}^i$ and assume it could not be reached through a branch in which some x_j , $j \in J_-^i$, was set to zero, that is,

$$x_j^r > 0 \quad \text{for all} \quad j \in J_-^i \ .$$

Then there will exist $\underset{\sim}{y}^i \geq \underset{\sim}{0}$, which is a convex linear combination of $\underset{\sim}{x}^r$ and $\underset{\sim}{x}^i$ which is feasible for P^i . As $L(\underset{\sim}{x}^i) < L(\underset{\sim}{x}^r)$, it follows that $L(\underset{\sim}{y}^i) < L(\underset{\sim}{x}^r)$.

So, by Lemma 2.1, there exists $\underset{\sim}{x}^s$, a descendant of $\underset{\sim}{x}^i$, for which $\underset{\sim}{x}^s \geq \underset{\sim}{0}$, and $L(\underset{\sim}{x}^s) \leq L(\underset{\sim}{y}^i) < L(\underset{\sim}{x}^r)$. Moreover, the method used in the proof of Lemma 2.1 only ever set $x_j^i < 0$ to zero in P^{i+1} . Hence a best feasible solution descendant from $\underset{\sim}{x}^i$ can be found by branching on only negative-valued variables at any node.

In the example given in Fig. 2.1 the tree generated using the pruning rule is shown by the thickened lines. The number of subproblems solved has been reduced from 32 to 11. However, tests done using this rule showed that the number of subproblems solved still rose exponentially with problem dimension. The main cause of the exponential rise in the work done appears to be the need to check all branches until the fathoming criteria are satisfied, to ensure the optimum has been found, although in each case tried the actual optimum was found early in the calculation.

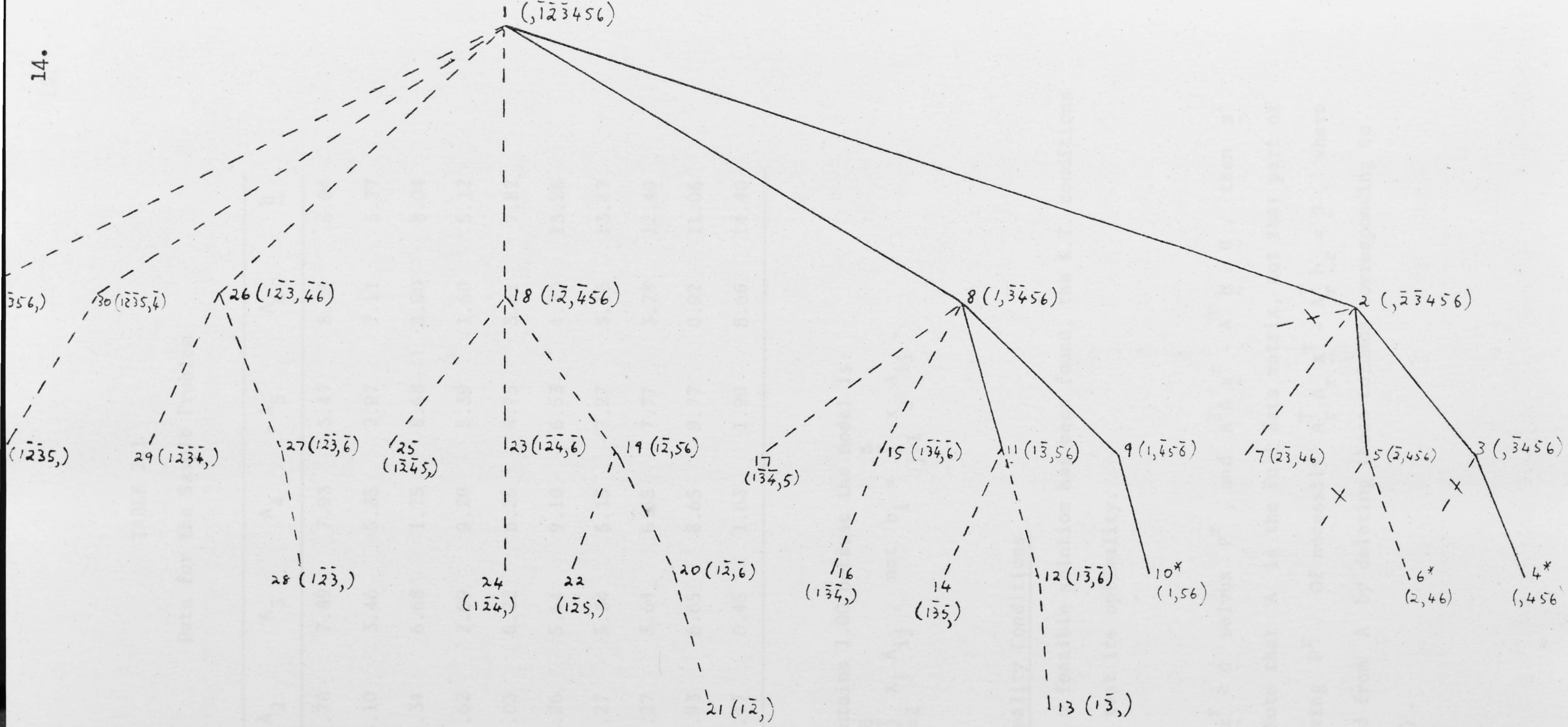


Figure 2.1 - A solution tree for the sample problem. At each node the numbers indicate which variables were used in the subproblem, those after the comma being optional. A "-" indicates that the variable is negative and a "*" indicates the solution is feasible (i.e. $x \geq 0$). The whole tree is generated by the Armstrong/Frome algorithm. The solid lines represent the tree generated using the improved pruning rule of theorem 2.1.

TABLE 2.1

Data for the Sample Problem

A_1	A_2	A_3	A_4	A_5	A_6	$\underset{\sim}{b}$
1.00	4.70	7.89	7.93	3.47	8.35	6.94
1.00	3.10	3.46	5.35	2.97	7.11	5.77
1.00	8.34	6.68	1.75	8.68	8.90	8.04
1.00	4.62	2.69	9.20	5.39	1.60	5.12
1.00	1.03	6.22	6.25	4.75	3.61	7.82
1.00	3.26	5.64	9.10	6.53	4.70	13.26
1.00	2.27	5.34	5.15	7.27	3.16	13.47
1.00	7.27	3.64	6.65	7.77	3.78	12.49
1.00	5.93	6.65	8.65	9.77	0.92	11.06
1.00	0.47	0.45	1.63	1.90	8.66	14.40

Column 1 contains 1.00 because the model is:

$b_i = x_1 + \sum_{j=2}^{6} x_j A_{ij}$, not $b_i = \sum_{j=1}^{6} x_j A_{ij}$.

2.2.3 Optimality Conditions

Once a feasible solution has been found, the K.T. conditions can be used to test its optimality.

Theorem 2.2

If $\underset{\sim}{x}^r \geq \underset{\sim}{0}$ solves P^r , and $A^T A \underset{\sim}{x}^r - A^T \underset{\sim}{b} \geq \underset{\sim}{0}$, then $\underset{\sim}{x}^r$ solves P . (Note that A is the full data matrix, not that part of it used in solving P^r . Of necessity $A_r^T A_r \underset{\sim}{x}^r - A_r^T \underset{\sim}{b}_r = \underset{\sim}{0}$, where A_r is obtained from A by deleting those columns corresponding to indices in J^r) .

TABLE 2.2

Solutions Obtained on Sample Program using Armstrong/Frome Algorithm

Node	x_1	x_2	x_3	x_4	x_5	x_6	$L(\underset{\sim}{x})$
1	-7.27	-1.89	-1.34	0.92	2.91	1.70	32.09
2	0	-1.52	-1.05	0.44	2.23	1.08	36.42
3	0	0	-0.95	0.49	1.21	0.84	102.00
4	0	0	0	0.25	0.84	0.62	127.18
5	0	-1.45	0	0.18	1.78	0.84	66.72
6	0	0.04	0	0.81	0	0.80	184.66
7	0	0.05	-0.02	0.82	0	0.81	184.65
8	10.22	0	-0.58	-0.20	0.59	0.04	86.07
9	13.31	0	0	-0.53	0.22	-0.30	93.51
10	7.52	0	0	0	0.33	0.08	103.49
11	8.10	0	-0.70	0	0.69	0.21	87.04
12	11.96	0	-0.35	0	0	-0.09	102.86
13	11.49	0	-0.34	0	0	0	103.45
14	9.73	0	-0.63	0	0.54	0	89.47
15	15.84	0	-0.17	-0.51	0	-0.40	94.45
16	12.34	0	-0.26	-0.20	0	0	100.99
17	10.67	0	-0.55	-0.23	0.56	0	86.11
18	4.43	-1.25	0	-0.07	1.44	0.50	64.25
19	3.57	-1.29	0	0	1.49	0.56	64.39
20	11.42	-0.29	0	0	0	-0.08	103.42
21	11.00	-0.28	0	0	0	0	103.91
22	8.29	-0.90	0	0	0.90	0	78.00
23	16.34	-0.24	0	-0.55	0	-0.42	92.22
24	12.39	-0.26	0	-0.24	0	0	99.92
25	10.11	-0.93	0	-0.37	1.00	0	68.90
26	16.37	-0.22	-0.08	-0.52	0	-0.41	92.01
27	12.38	-0.20	-0.26	0	0	-0.09	100.75
28	11.90	-0.20	-0.25	0	0	0	101.36
29	12.78	-0.21	-0.17	-0.20	0	0	98.82
30	10.83	-0.88	-0.47	-0.28	1.14	0	61.56
31	9.69	-0.85	-0.57	0	1.10	0	66.50
32	4.05	-1.34	-0.76	0	1.93	0.73	45.04

Proof

The K.T. conditions for the MPP (2.3) are stated in (2.4), (2.5) and (2.6). Here, we have

$$g(\underset{\sim}{x}) = \underset{\sim}{x}$$

$$\nabla g(\underset{\sim}{x}) = I \quad \text{and} ,$$

$$\nabla L(\underset{\sim}{x}) = 2(A^TA\underset{\sim}{x} - A^T\underset{\sim}{b}) .$$

Let $\underset{\sim}{\lambda}^r = \nabla L(\underset{\sim}{x}^r) \geq \underset{\sim}{0}$; then

$$\nabla L(\underset{\sim}{x}^r) = \underset{\sim}{\lambda}^r = \nabla g(\underset{\sim}{x}^r)\, \underset{\sim}{\lambda}^r .$$

Also, as $\underset{\sim}{x}^r$ solves P^r ,

$$\lambda_i^r = \nabla L(\underset{\sim}{x}^r)_i = 0 \quad \text{for } i \notin J^r ,$$

but

$$x_i^r = 0 \quad \text{for } i \in J^r ;$$

hence

$$\underset{\sim}{\lambda}^{rT} \underset{\sim}{x}^r = 0 .$$

So $(\underset{\sim}{x}^r, \underset{\sim}{\lambda}^r)$ is a K.T. point for P . Hence $\underset{\sim}{x}^r$ solves P .

2.2.4 Selection of the Next Node

If the above test for optimality fails, it can still be used to determine the next node to branch on, and which branching variable should be chosen at that node.

Theorem 2.3

If $\underset{\sim}{x}^r \geq \underset{\sim}{0}$ solves P^r , $\underset{\sim}{\lambda}^r = \nabla L(\underset{\sim}{x}^r)$, $\lambda_k^r < 0$ for some $k \in J^r$, and $J^{r+1} = \{i | x_i^r = 0, i \neq k\}$, then there is some descendant, $\underset{\sim}{x}^s$, of $\underset{\sim}{x}^{r+1}$ which is feasible and for which $L(\underset{\sim}{x}^s) < L(\underset{\sim}{x}^r)$.

Proof

Using the well-known convex function property

$$f(\underset{\sim}{x}_2) \geq f(\underset{\sim}{x}_1) + \nabla f(\underset{\sim}{x}_1)^T(\underset{\sim}{x}_2 - \underset{\sim}{x}_1) ,$$

we have $L(\underset{\sim}{x}^{r+1}) \geq L(\underset{\sim}{x}^r) + \underset{\sim}{\lambda}^{rT}(\underset{\sim}{x}^{r+1} - \underset{\sim}{x}^r)$.

But $L(\underset{\sim}{x}^{r+1}) < L(\underset{\sim}{x}^r)$, as either P^{r+1} is less restricted than $P^r (J^{r+1} \subset J^r)$, or else, if $x_i^r = 0$ for some $i \notin J^r$, then $\underset{\sim}{x}^r$ solves $P^{r'}$, where $J^{r'} = J^r \cup \{i\}$, and again $J^{r+1} \subset J^{r'}$. Thus

$$0 > \underset{\sim}{\lambda}^{rT}(\underset{\sim}{x}^{r+1} - \underset{\sim}{x}^r) = \underset{\sim}{\lambda}^{rT} \underset{\sim}{x}^{r+1} = \lambda_k^r x_k^{r+1} .$$

Hence

$$x_k^{r+1} > 0 .$$

Hence there exists $\underset{\sim}{y}^{r+1} \geq \underset{\sim}{0}$ which is a convex linear combination of $\underset{\sim}{x}^{r+1}$ and $\underset{\sim}{x}^r$, and so $L(\underset{\sim}{y}^{r+1}) < L(\underset{\sim}{x}^r)$, and which is feasible for P^{r+1} . The proof now follows from Lemma 2.1.

One point worth noting is the definition of J^{r+1} above. It could not be defined as $\{j | j \in J^r, j \neq k\}$, as it may be that $x_i^r = 0$ for some $i \notin J^r$ and, if $x_i^{r+1} < 0$, then no convex linear combination, $\underset{\sim}{y}^{r+1}$ of $\underset{\sim}{x}^r$ and $\underset{\sim}{x}^{r+1}$ can have $y_i^{r+1} \geq 0$.

Figure 2.2 shows that portion of the tree generated using Theorems 2.2 and 2.3 on the test problem of Fig. 2.1. The nodes generated are given in Table 2.3. The first four nodes correspond to nodes 1 to 4 of the earlier tree, and nodes 5 and 6 correspond to nodes 9 and 10 respectively. In terms of the primal/dual feasibility discussion of Section 2.1.1, nodes 1 and 2 are dual feasible, node 3 is neither dual nor primal feasible, node 4 is primal feasible, node 5 is dual feasible and node 6, the optimum, is, of course, both.

It should be mentioned here that the example was chosen for its illustrative properties rather than its typicality. In over 90% of the test problems solved, the first (primal) feasible solution found was the optimum, and in the majority of the remainder, the algorithm jumped directly to the optimum.

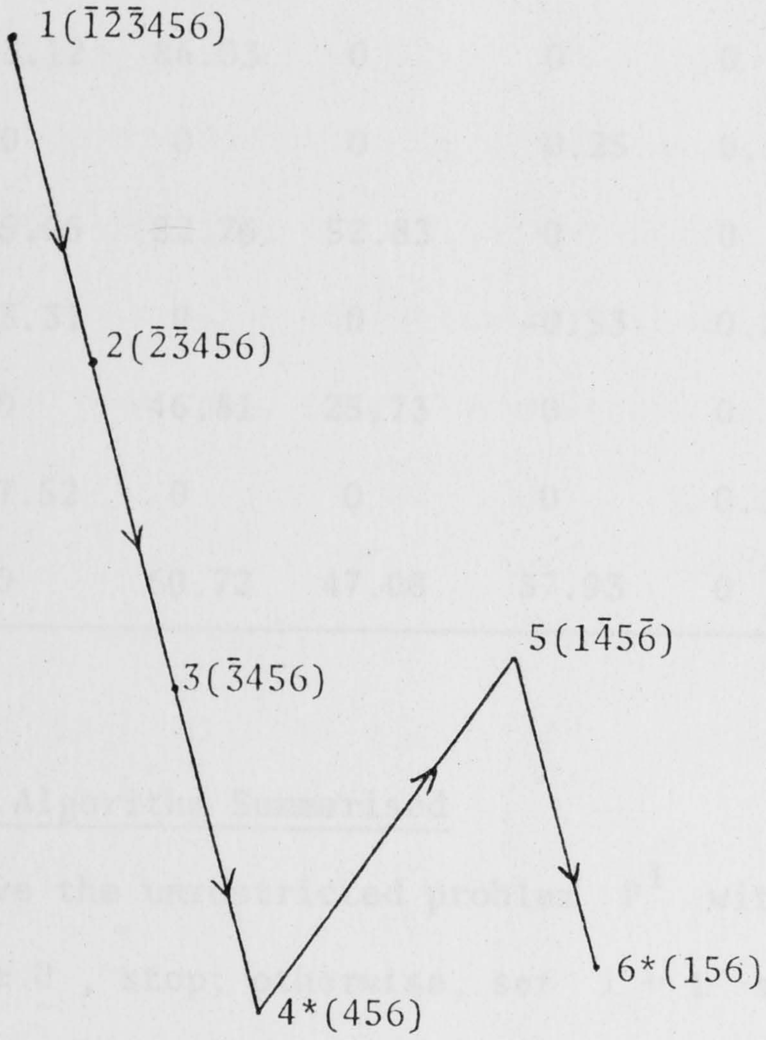


FIG 2.2 . The solution tree for the sample problem using the new algorithm. The numbers in parentheses represent the variables used in the solution, those with a -, being negative. An * indicates that the solution is feasible (that is, $\underset{\sim}{x} \geq \underset{\sim}{0}$).

TABLE 2.3

Solutions obtained on the Sample Problem using the New Algorithm

Node	x_1/λ_1	x_2/λ_2	x_3/λ_3	x_4/λ_4	x_5/λ_5	x_6/λ_6	$L(\underset{\sim}{x})$
1	-7.27	-1.89	-1.34	0.92	2.91	1.70	32.09
	0	0	0	0	0	0	
2	0	-1.52	-1.05	0.44	2.23	1.08	36.42
	1.19	0	0	0	0	0	
3	0	0	-0.95	0.49	1.21	0.84	102.00
	-3.12	86.03	0	0	0	0	
4	0	0	0	0.25	0.84	0.62	127.18
	-5.06	82.76	52.83	0	0	0	
5	13.31	0	0	-0.53	0.22	-0.30	93.51
	0	46.81	25.73	0	0	0	
6	7.52	0	0	0	0.33	0.08	103.49
	0	60.72	47.08	37.93	0	0	

2.2.5 The Algorithm Summarised

1. Solve the unrestricted problem P^1 with $J^1 = \emptyset$. If $\underset{\sim}{x}^1 \geq \underset{\sim}{0}$, stop; otherwise, set $i \leftarrow 1$ and go to 2.

2. Set $i \leftarrow i+1$. Use some heuristic to select k from the set $\{j|x_j^{i-1} < 0\}$ and let $J^i = J^i \cup \{k\}$. Solve P^i . If $\underset{\sim}{x}^i \geq \underset{\sim}{0}$, go to 3; otherwise, go to 2.

3. Use Theorem 2.2 to test $\underset{\sim}{x}^i$ for optimality. If the test succeeds stop; otherwise, go to 4.

4. Use some heuristic to choose k from the set $\{j|\lambda_j^i < 0\}$ where $\underset{\sim}{\lambda}^i = \nabla L(\underset{\sim}{x}^i)$; set $i \leftarrow i+1$, and let $J^i = \{j|x_j^{i-1} = 0, j \neq k\}$. Solve P^i . If $\underset{\sim}{x}^i \geq \underset{\sim}{0}$, go to 3;

otherwise, go to 5.

5. Set $i \leftarrow i + 1$. Choose k according to the method used in the proof of Lemma 2.1. Let $J^i = J^{i-1} \cup \{k\}$. Solve P^i. If $\underset{\sim}{x}^i \geq \underset{\sim}{0}$, go to 3; otherwise, go to 5.

The heuristics used in steps 2 and 4 were to choose the most negative variable in each case (but see Section 2.4 for a fuller discussion).

2.3 COMPLEXITY

It appears difficult to determine any absolute complexity bounds for the algorithm, but if the assumption is made that, at any feasible $\underset{\sim}{x}^r$ which fails the optimality test of Theorem 2.2, it does so for only one variable, then linear bounds can be derived.

Theorem 2.4

If $\underset{\sim}{x}^r \geq \underset{\sim}{0}$ solves P^r, $\underset{\sim}{\lambda}^r = \nabla L(\underset{\sim}{x}^r)$, and $\lambda_i^r \geq 0$ for $i \in J^r - \{k\}$, then N, the number of subproblems solved, is at most $2n$.

Proof

Let

$$J^{r+i} = J^r - \{i\} \quad \text{for all } i \in J^r - \{k\} .$$

Then, by an argument similar to that used in proving Theorem 2.3,

$$x_i^{r+i} < 0 \quad \text{for all } i \in J^r - \{k\} .$$

Now for any $\underset{\sim}{x}^s \geq \underset{\sim}{0}$ such that $L(\underset{\sim}{x}^s) < L(\underset{\sim}{x}^r)$, assume that $x_k^s = 0$. Let

$$J' = \{j \mid x_j^s > 0 , \; j \in J^r\} .$$

Then there will exist a convex linear combination, $\underset{\sim}{y}^r \geq \underset{\sim}{0}$, of $\underset{\sim}{x}^r$, $\underset{\sim}{x}^s$ and $\underset{\sim}{x}^{r+i}$, $i \in J'$, which is feasible for P^r and for which $L(\underset{\sim}{y}^r) < L(\underset{\sim}{x}^r)$, which contradicts the optimality of $\underset{\sim}{x}^r$. Hence for each subsequent feasible solution, $\underset{\sim}{x}^s$, found after $\underset{\sim}{x}^r$, $x_k^s > 0$.

Now this applies at each step, so that after each feasible solution is found by the algorithm, one more variable must remain strictly positive. Thus, if α_i is the number held to zero in the ith feasible solution found, then $\alpha_i \leq n + 1 - i$, and also there will be at most n feasible solutions found.

Let f_i be the number of subproblems solved between the (i - 1)st (exclusive) and ith (inclusive) feasible subproblems, and F the total number of feasible solutions found. Evidently, for $i > 1$, $f_i = \alpha_i - \alpha_{i-1} + 2$, and if α_0 is defined as 1, the formula is also correct for f_1.

Thus

$$\begin{aligned} N &= \sum_{i=1}^{F} f_i \\ &= \sum_{i=1}^{F} \alpha_i - \alpha_{i-1} + 2 \\ &= 2F + \alpha_F - \alpha_0 \\ &\leq 2F + n + 1 - F - 1 \\ &\leq 2n \; . \end{aligned}$$

Although there are no a priori grounds for supposing that the above assumptions are always true, no case has yet been found in which the assumption did not hold. Indeed, the example of Fig. 2.1, with $N = n$, was the only instance of $N/n \geq 1$ (see Table 2.4).

2.4 RESULTS

The algorithm was tested against data generated using a random number generator. For each problem size, ten sets of data were solved. Due to lack of consistency of execution times, N, the number of subproblems solved, was taken as the measure of algorithm efficiency. (As an indication, however, solving problems with 40 variables and 50 rows of data took 5 to 10 seconds on a Univac 1110/42.) Armstrong and Frome claimed competitiveness for their algorithm, and as it was the progenitor of the new algorithm, it was used for comparison purposes. The results, given in Table 2.4, display the linearity predicted in Section 2.3.

Of the several heuristics tested for choosing the variable to be set to zero at step 2 of the algorithm, choosing the most negative and choosing the negative variable which had differed least from $\underset{\sim}{x}^1$ proved best. The former was chosen for its simplicity. No choice ever had to be made at step 4, but choosing the most negative λ_i is suggested.

2.5 EXTENSION TO OTHER PROBLEMS

The features of the restricted least squares problem which make it suitable for the algorithm as given are:

(i) the strict convexity of the objective function;

(ii) the special nature of the constraints; and

(iii) the ease with which each subproblem P^i can be solved.

Any problem which has the above properties is suitable for solving by the algorithm. One question which arises in other applications is the optimality test if the Kuhn-Tucker conditions are not readily available, as this test is central to the algorithm. If this is the case, Theorem 2.2 can be replaced by one which requires solving no more than $n-1$ additional subproblems to test the optimality of a feasible solution to a subproblem.

<u>Theorem 2.5</u>

Let $\underset{\sim}{x}^r \geq \underset{\sim}{0}$ solve P^r . Define $J^{r+i} = J^r - \{i\}$ for all $i \in J^r$. If $n_i^{r+i} < 0$ for all $i \in J^r$, then $\underset{\sim}{x}^r$ solves P .

<u>Proof</u>

Assume the above conditions hold and further assume that there exists $\underset{\sim}{x}'$ such that $L(\underset{\sim}{x}') < L(\underset{\sim}{x}^r)$. Now, for $i \in J^r$, $x_i' \geq 0$, and for some $i \in J^r$, $x_i' > 0$. But, for $i \in J^r$, $x_i^{r+i} < 0$. Hence there is a convex linear combination, $\underset{\sim}{x}^s$, of $\underset{\sim}{x}'$ and $\underset{\sim}{x}^{r+i}$ for all $i \in J^r$, for which $x_i^s = 0$ for all $i \in J^r$, so that $\underset{\sim}{x}^s$ is feasible for P^r .

Now $L(\underset{\sim}{x}^s) \leq$ convex linear combination $(L(\underset{\sim}{x}')$, $L(\underset{\sim}{x}^{r+i})$ for all $i \in J^r)$. Since $L(\underset{\sim}{x}') > L(\underset{\sim}{x}^r)$, $L(\underset{\sim}{x}^{r+1}) \leq L(\underset{\sim}{x}^r)$ and the contribution to $\underset{\sim}{x}'$ of $\underset{\sim}{x}^s$ is non-zero, it follows that $L(\underset{\sim}{x}^s) < L(\underset{\sim}{x}^r)$, which contradicts the assumption that $\underset{\sim}{x}^r$ solves P^r . Hence $\underset{\sim}{x}^r$ solves P .

The only modifications to the algorithm necessary are to replace Theorem 2.2 with Theorem 2.5 in step 3, and to omit step 4.

Under the assumptions of Theorem 2.4 (the complexity theorem), it is easy to show that the number of subproblems solved is not more than $\frac{1}{2}n(n+1)$. However, testing the modified algorithm on the same test data as used before indicated that in practice this algorithm also behaves linearly (see Table 2.4).

2.6 CONCLUSION

The algorithm presented here appears to be a considerable improvement on existing algorithms of branch and bound type for this problem, without sacrificing any of the advantages of these algorithms, for example, ease of use in an interactive mode, wide availability of least squares regression routines, and simple modification to account for variable bounds.

The reason would appear to be that in this problem, as so often in branch and bound, the optimum solution is found quickly and then much time is spent in the subsequent searching necessary to verify it. Thus the biggest advantage of this approach is in the use of optimality conditions to improve bounding. Incorporating this into the general framework of branch and bound has resulted in a very efficient algorithm.

TABLE 2.4

Experimental results. Number of subproblems solved

Dimension of A		Armstrong/Frome algorithm		Armstrong/Frome improved		New Algorithm		New Algorithm modified	
n	m	Mean	Worst	Mean	Worst	Mean	Worst	Mean	Worst
6	10	10.0	32	4.4	11	3.3	6	4.9	13
10	15	168.2	566	34.6	96	5.4	8	8.8	14
15	20	4097.8	11886	668.6	1947	10.0	13	18.0	24
20	30	—	—	—	—	11.9	14	21.8	26
30	40	—	—	—	—	16.6	19	31.2	36
40	50	—	—	—	—	24.4	28	46.8	54

CHAPTER 3

AN EFFICIENT IMPLEMENTATION OF THE LEAST SQUARES ALGORITHM

3.1 INTRODUCTION

In the previous chapter we presented an algorithm for solving the restricted least squares problem

$$(3.1) \qquad \text{minimise} \quad \| A^T \underset{\sim}{x} - \underset{\sim}{b} \| \qquad \text{subject to} \quad \underset{\sim}{x} \geq \underset{\sim}{0} \; .$$

In the original implementation of that algorithm, standard regression routines were used to solve a new subproblem at each iteration. Now although the algorithm appeared efficient in terms of the number of subproblems solved (i.e. number of nodes of the search tree to be visited), the implementation of the algorithm is still not good. In this chapter we consider an improved implementation of the algorithm. In particular, we want to provide methods which avoid solving each of the unconstrained least squares problems ab initio when only one of the variables is changed at each step. The key to our approach is suggested by the Kuhn-Tucker conditions which characterize the unique minimum of (3.1), with uniqueness following from the strict convexity of the objective function and the linearity of the constraints. These conditions are

$$(3.2a) \qquad -A^T(\underset{\sim}{b} - A\underset{\sim}{x}) = \underset{\sim}{\lambda} \; ,$$

$$(3.2b) \qquad \underset{\sim}{\lambda} \geq \underset{\sim}{0} \; , \; \underset{\sim}{x} \geq \underset{\sim}{0} \; ,$$

and

(3.2c) $\lambda_i x_i = 0\ , \quad i = 1,2,\ldots,n\ ,$

so that the subset selection problem can be restated as that of seeking among all solutions satisfying the system of equations (3.2a) and the complementarity condition (3.2c) the unique pair satisfying $\underset{\sim}{\lambda} \geq \underset{\sim}{0}\ ,\ \underset{\sim}{x} \geq \underset{\sim}{0}$.

From our point of view the striking feature of this formulation is the symmetry between the roles of $\underset{\sim}{x}$ and $\underset{\sim}{\lambda}$. In particular it is possible to interchange the roles of $\underset{\sim}{x}$ and $\underset{\sim}{\lambda}$ in the algorithm. This has the advantage that a certain amount of initial processing is avoided. For example, starting with $\underset{\sim}{x}$ as the unconstrained minimizer of (3.1), is equivalent to rewriting (3.2a) in the form

(3.3) $$(A^TA)^{-1}\underset{\sim}{\lambda} - \underset{\sim}{x} = -(A^TA)^{-1}\ A^T\underset{\sim}{b}$$

and satisfying (3.2c) by setting $\underset{\sim}{\lambda} = \underset{\sim}{0}$.

By the complementarity condition (3.2c), fixing a particular component of $\underset{\sim}{x}$ at zero is equivalent to freeing the corresponding component of $\underset{\sim}{\lambda}$. Thus we consider at each stage a partition of $\underset{\sim}{x}$,

(3.4a) $$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{x}_1 \\ \underset{\sim}{x}_2 \end{bmatrix} \text{ with } \underset{\sim}{x}_2 = \underset{\sim}{0}\ ,$$

and a corresponding partition of $\underset{\sim}{\lambda}$,

(3.4b) $$\underset{\sim}{\lambda} = \begin{bmatrix} \underset{\sim}{\lambda}_1 \\ \underset{\sim}{\lambda}_2 \end{bmatrix} \text{ with } \underset{\sim}{\lambda}_1 = \underset{\sim}{0}\ ,$$

which ensure automatically that (3.2c) is satisfied. If (3.2a) is now solved for the variables permitted to be nonzero at the current stage then it can be written

(3.5) $$\underset{\sim}{\sigma}_A - M\underset{\sim}{\sigma}_1 = -\underset{\sim}{q}\ ,$$

where

$$(3.6) \qquad \underset{\sim}{\sigma}_A = \begin{bmatrix} \underset{\sim}{x}_1 \\ \underset{\sim}{\lambda}_2 \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\sigma}_1 = \begin{bmatrix} \underset{\sim}{\lambda}_1 \\ \underset{\sim}{x}_2 \end{bmatrix} .$$

Each step of the algorithm involves interchanging a component of $\underset{\sim}{x}$ with the corresponding component of $\underset{\sim}{\lambda}$. This results in a transformation of (3.5) which can be represented by multiplication by an elementary Jordan matrix followed by appropriate permutations to partition the new variables into the form (3.6). We define the Jordan matrix J_i by

$$(3.7) \qquad J_i K_i(M) = (I-\underset{\sim}{j}_i \underset{\sim}{e}_i^T) K_i(M) = -\underset{\sim}{e}_i ,$$

where i is the index of the element of $\underset{\sim}{\sigma}_1$ to be exchanged, and $K_i(.)$ indicates that the ith column is taken. Using a bar to denote transformed quantities we have

$$\bar{\underset{\sim}{q}} = \underset{\sim}{q} - q_i \underset{\sim}{j}_i ,$$

so

$$\left.\begin{aligned} &(3.8a) \qquad \bar{q}_k = q_k - \frac{q_i M_{ik}}{M_{ii}} , \quad k \neq i , \\ &\text{and} \qquad \bar{q}_i = -\frac{q_i}{M_{ii}} , \end{aligned}\right\}$$

$$K_k(\bar{M}) = (I-\underset{\sim}{j}_i \underset{\sim}{e}_i^T) K_k(M)$$

$$(3.8b) \qquad = K_k(M) - \frac{M_{ik}}{M_{ii}} \{K_i(M)+\underset{\sim}{e}_i\} , \quad k \neq i$$

When $k = i$, the ith column of $\bar{M}$ comes as a result of the interchange $\lambda_i \leftrightarrow x_i$. This gives

$$(3.8c) \qquad K_i(\bar{M}) = -(I-\underset{\sim}{j}_i \underset{\sim}{e}_i^T)\underset{\sim}{e}_i$$

$$= \frac{1}{M_{ii}} \{K_i(M)+\underset{\sim}{e}_i\} - \underset{\sim}{e}_i .$$

This shows that the computations involved in the algorithm can be carried out in a manner familiar from stepwise regression (see eg, Effroymsom, 1960). However, the problem set-up still involves the calculation of the normal matrix which is a significant initial computation.

An alternative to forming the normal matrix is to apply orthogonal transformations to the data matrix. This is known to have superior numerical properties (see eg, Golub and Wilkinson, 1966) but it is interesting that in the stepwise regression case it is known to be more efficient for an important range of values of m and n (Osborne, 1976). This approach is considered in the next section. It turns out that set-up time can be considerably reduced by working with the multiplier vector $\underset{\sim}{\lambda}$, and there is an unexpected bonus for $\underset{\sim}{\lambda}_2$ turns out to be a numerically better determined quantity than $\underset{\sim}{x}_1$. Numerical results, including a comparison with the quadratic programming approach, are presented in Section 3.3.

3.2 USE OF ORTHOGONAL TRANSFORMATIONS

To derive the equations satisfied by $\underset{\sim}{x}_1$ and $\underset{\sim}{\lambda}_2$ we assume that the orthogonal transformation of the data matrix is given by

$$(3.9) \qquad A = Q \begin{bmatrix} U \\ 0 \end{bmatrix} \quad \text{and} \quad Q^T\underset{\sim}{b} = \begin{bmatrix} \underset{\sim}{c}_1 \\ \underset{\sim}{c}_2 \end{bmatrix} ,$$

where Q is orthogonal and U upper triangular. Substituting in (3.2a) gives

$$\underset{\sim}{\lambda} - U^TU\underset{\sim}{x} = -U^T\underset{\sim}{c}_1$$

or

$$(3.10) \qquad U^{-T}\underset{\sim}{\lambda} - U\underset{\sim}{x} = -\underset{\sim}{c}_1 \; .$$

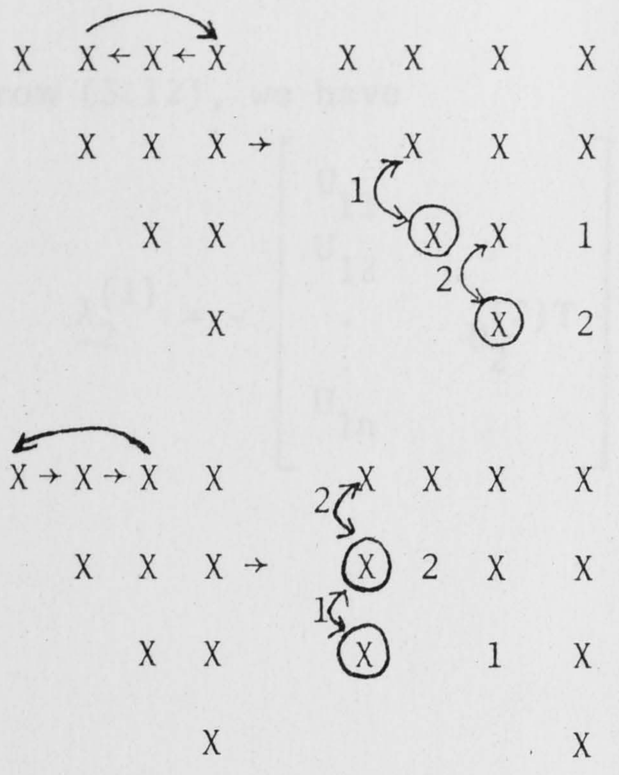
We partition U and $\underset{\sim}{c}_1$ to conform with (3.4) by setting

$$(3.11) \qquad U = \begin{bmatrix} U_1 & U_{12} \\ 0 & U_2 \end{bmatrix} \quad \text{and} \quad \underset{\sim}{c}_1 = \begin{bmatrix} \underset{\sim}{c}_{11} \\ \underset{\sim}{c}_{12} \end{bmatrix}$$

so that (3.10) reduces to the pair of equations

$$(3.12) \qquad U_1\underset{\sim}{x}_1 = \underset{\sim}{c}_{11} \quad \text{and} \quad \underset{\sim}{\lambda}_2 = -U_2^T\underset{\sim}{c}_{12} \; .$$

The interchange of a pair λ_i , x_i destroys the form of U unless the last element of $\underset{\sim}{x}_1$ becomes the first element of $\underset{\sim}{\lambda}_2$ or vice versa. Thus the upper triangular form of U must be restored following an interchange, and this can be done using the now standard techniques treated in detail by Gill, et al, 1974. For example, to drop the kth element of $\underset{\sim}{x}_1$ which we assume to be of length $p > k$, we perform the interchanges $k + 1 \rightarrow k$, $k + 2 \rightarrow k + 1, \ldots, k \rightarrow p$ on the columns of U_1 and then sweep out the elements introduced in the sub-diagonal positions using plane rotations $W\{j,j+1,(j+1,j)\}$, $j = k,k + 1,\ldots,p - 1$ where $W\{i,j,(p,q)\}$ is the plane rotation mixing rows i and j and making zero the element in the (p,q) position. Similarly, to add an element to $\underset{\sim}{x}_1$ the corresponding column (say k) of U_2 is moved to column 1 by the sequence of interchanges $1 \rightarrow 2$, $2 \rightarrow 3, \ldots, k \rightarrow 1$, and the upper triangular form is restored by the sequence of plane rotations $W\{j,j+1,(j+1,1)\}$, $j = k - 1,\ldots,1$. These operations are shown schematically in Fig. 3.1. The interchanges are indicated by arrows, elements eliminated are circled, and elements introduced are labelled by the rotation number.



deletion of variable from $\underset{\sim}{x}_1$ (matrix shown is U_1)

addition of variables to $\underset{\sim}{x}_1$ (matrix shown is U_2)

Fig 3.1 Transformations for addition and deletion of variables.

The algorithm can now proceed as before. However, although it appears from the above description that the initial set up time includes the factorization (3.9), the observation that it is possible to work with $\underset{\sim}{\lambda}$ instead of $\underset{\sim}{x}$ makes it possible to start the algorithm without any pre-processing of the data matrix A. The key point is that $\underset{\sim}{\lambda}_2$ can be determined once the transformation necessary for the calculation of the complementary set $\underset{\sim}{x}_1$ has been carried out, although $\underset{\sim}{x}_1$ need not be computed unless $\underset{\sim}{\lambda}_2 \geq \underset{\sim}{0}$. The modification to the algorithm is explained by considering the first step which is typical. Note that initially $\underset{\sim}{x} = \underset{\sim}{x}_2^{(1)} = \underset{\sim}{0}$ so that (3.2a) gives

$$(3.13) \qquad \underset{\sim}{\lambda}_2^{(1)} = -U^T\underset{\sim}{c}_1 = -A^T\underset{\sim}{b} ,$$

where the superscript indicates step number. Using a Householder transformation (say) to sweep out the first column of A gives

$$H_1A = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ 0 & X & \cdots & X \\ \vdots & \vdots & & \vdots \\ 0 & X & \cdots & X \end{bmatrix} \qquad \text{and} \qquad H_1\underset{\sim}{b} = \begin{bmatrix} c_1 \\ X \\ \vdots \\ X \end{bmatrix}$$

Now, from (3.12), we have

$$\underset{\sim}{\lambda}_2^{(1)} = - \left[\begin{array}{c|c} U_{11} & \\ U_{12} & \\ \vdots & U_2^{(2)T} \\ U_{1n} & \end{array}\right] \begin{bmatrix} c_1 \\ \underset{\sim}{c}_{12}^{(2)} \end{bmatrix}$$

$$= -c_1 \begin{bmatrix} U_{11} \\ U_{12} \\ \vdots \\ U_{1n} \end{bmatrix} + \begin{bmatrix} 0 \\ \underset{\sim}{\lambda}_2^{(2)} \end{bmatrix} , \tag{3.14}$$

showing that the Lagrange multipliers can be updated and decisions made on the order in which the remaining columns of A are swept out as the factorization of A proceeds. Essentially no set-up computations are required for this form of the algorithm.

This relation can be given a general form. We partition Q so that (3.9) is written

$$\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} A = \begin{bmatrix} U \\ 0 \end{bmatrix} \quad \text{and} \quad Q_1^T \underset{\sim}{b} = \underset{\sim}{c}_1 . \tag{3.15}$$

Partitioning A and Q_1^T in conformity with $\underset{\sim}{x}$ we obtain

$$Q_{11}^T [A_1 \ A_2] = [U_1 \ U_{12}] \quad \text{and} \quad Q_{12}^T [A_1 \ A_2] = [0 \ U_2] , \tag{3.16}$$

so that

$$-\underset{\sim}{\lambda}_2 = U_2^T \underset{\sim}{c}_{12} = A_2^T Q_{12} Q_{12}^T \underset{\sim}{b}$$

and, using (3.16),

$$\begin{aligned} -\begin{bmatrix} \underset{\sim}{0} \\ \underset{\sim}{\lambda}_2 \end{bmatrix} &= A^T Q_{12} Q_{12}^T \underset{\sim}{b} \\ &= A^T[I - Q_{11} Q_{11}^T]\underset{\sim}{b} \,, \end{aligned} \tag{3.17}$$

as

$$Q_{12} Q_{12}^T = I - Q_{11} Q_{11}^T - Q_2 Q_2^T \quad \text{and} \quad A^T Q_2 = 0 \,.$$

In particular, the general form for (3.14) is

$$-\underset{\sim}{\lambda}_2 = A_2^T \underset{\sim}{b} - U_{12}^T \underset{\sim}{c}_{11} \,, \tag{3.18}$$

and this confirms that the multiplier vector is available when only the transformation of A necessary to compute $\underset{\sim}{x}_1$ has been completed.

Equation (3.18) is useful also as it permits an error analysis for this method of computing $\underset{\sim}{\lambda}_2$ to be given. Indicating computed quantities by bars we have

$$\begin{aligned} \overline{\underset{\sim}{\lambda}}_2 - \underset{\sim}{\lambda}_2 = \delta\underset{\sim}{\lambda} &= \overline{U}_{12}^T \overline{\underset{\sim}{c}}_{11} - A_2^T Q_{11} Q_{11}^T \underset{\sim}{b} + \underset{\sim}{\varepsilon} \\ &= \{\overline{U}_{12}^T - A_2^T Q_{11}\}\overline{\underset{\sim}{c}}_{11} + A_2^T Q_{11}\{\overline{\underset{\sim}{c}}_{11} - \underset{\sim}{c}_{11}\} + \underset{\sim}{\varepsilon} \,, \end{aligned} \tag{3.19}$$

where $\underset{\sim}{\varepsilon}$ is the evaluation error. This equation can be further expanded to give

$$\begin{aligned} \delta\underset{\sim}{\lambda} = \{\overline{U}_{12}^T - A_2^T Q'_{11}\}\overline{\underset{\sim}{c}}_{11} &+ A_2^T\{Q'_{11} - Q_{11}\}\overline{\underset{\sim}{c}}_{11} \\ &+ A_2^T Q_{11}\{Q'^T_{11} - Q_{11}^T\}\underset{\sim}{b} + A_2^T Q_{11}\{\overline{Q}_{11}^T - Q'^T_{11}\}\underset{\sim}{b} + \underset{\sim}{\varepsilon} \,, \end{aligned} \tag{3.20}$$

where the prime indicates the exact orthogonal factorization defined by the actual numeric data at each stage. The quantities on the right-hand side of (3.20) can now be estimated using known inequalities.

The most important terms are those involving $Q'_{11} - Q_{11}$, and it was shown by Jennings and Osborne (1974) that this can be bounded by an expression of the form $k_1 \text{eps}\,\chi(A_1)$ where k_1 is a constant, eps is the machine precision, and $\chi(A_1)$ is the spectral condition number of A_1. This is a result which is more favourable than the corresponding result for $\underset{\sim}{x}_1$ which Golub and Wilkinson (1966) showed to have a dependence also on

$$\text{eps}\chi(A_1)^2 \left\| \begin{matrix} \underset{\sim}{c}_{12} \\ \underset{\sim}{c}_2 \end{matrix} \right\| .$$

However, the bounds quoted in the error estimate are for the usual form of orthogonal factorization which takes no account of the possibility of the back-tracking which can and does occur in the algorithm. If we assume the analysis is valid also in the case of back-tracking, then presumably we have to use the largest condition number encountered to the present stage rather than the condition number of the current partition A_1.

One further point in favour of this form of the algorithm is that it appears rarely to be necessary to compute the complete factorization (3.9) in the determination of the optimum subset $\underset{\sim}{x}_1$. It is conceivable that the full system could be badly conditioned while the subproblems leading to the optimal subset could be well conditioned.

3.3 NUMERICAL RESULTS

A subset selection algorithm for (3.1) proceeds essentially in two stages: an initial search for a feasible $\underset{\sim}{x}_1$ (or $\underset{\sim}{\lambda}_2$) using a heuristic to determine at each stage the component to be set to zero, and subsequent back-tracking to explore other branches of the search tree if the first feasible solution is not optimal. It is in this second phase of the computation that the major improvements due to the algorithm are achieved. In the first phase the heuristic commonly used is to fix at zero level the most negative of the current solution components. This procedure suffers from the disadvantage that it is not scale independent. For example, if the data matrix A is multiplied by a diagonal matrix D to rescale the column norms so that

$$(3.21a) \qquad A \leftarrow AD^{-1} \; ,$$

then it follows from (3.2a) that the solution vectors are transformed by

$$(3.21b) \qquad \underset{\sim}{x} \leftarrow D\underset{\sim}{x} \quad \text{and} \quad \underset{\sim}{\lambda} \leftarrow D^{-1}\underset{\sim}{\lambda} \; .$$

Our numerical experiments have shown that the choice of the first phase heuristic is important because it can affect the amount of work that has to be done in the second phase of the computation. It seems reasonable that a good heuristic should not be affected by changes in scale, and for this reason we compare the choice of most negative component with a choice which corresponds to the test used in stepwise regression to determine the variable to enter the regression at each step and which has the property of invariance with respect to column scaling. If we consider the data matrix factorized so that at the ith step of the first phase of the computation we have

$$(3.22) \qquad A^{(i)} = \begin{bmatrix} U_1 & U_{12} \\ 0 & B \end{bmatrix} \quad \text{and} \quad \underset{\sim}{b}^{(i)} = \begin{bmatrix} \underset{\sim}{c}_{11} \\ \underset{\sim}{d} \end{bmatrix} ,$$

then the stepwise regression test selects the variable to be introduced as that which maximizes

$$|\underset{\sim}{d}^T K_j(B)|/\|K_j(B)\| ,$$

as this leads to the biggest reduction in the sum of squares of the residuals (see Golub, 1965). Here $\|K_j(B)\|$ is the euclidean length of the jth column of B . Now, from (3.18),

$$(3.23) \qquad -\underset{\sim}{\lambda}_2^{(i)} = A_2^T \underset{\sim}{b} - U_{12} \underset{\sim}{c}_{11} = B^T \underset{\sim}{d} ,$$

so that we can use the stepwise test in the form

$$(3.24) \qquad \underset{\sim}{x}_1 \leftarrow \begin{bmatrix} \underset{\sim}{x}_1^{(i)} \\ x_s \end{bmatrix} \quad s \text{ maximizes } \gamma(j) = -\frac{(\underset{\sim}{\lambda}_2^{(i)})_j}{\|K_j(B)\|}$$

for all j such that $\gamma(j) > 0$.

However, our implementation actually considers $\gamma(j)^2$ as $\|K_j(B)\|^2$ is readily updated from step to step.

We report numerical results for two sets each of ten problems with $m = 50$, $n = 40$. The data are obtained by sampling from a normal distribution for the first set and from a uniform distribution for the second set, except that in all cases $K_1(A)_j = 1$, $j = 1,2,\ldots,m$. For each set we give results for each of the selection strategies already discussed and for the case in which the most negative strategy is used after the columns of A are scaled initially to have unit length. Also, for the data drawn from the uniform distribution, we consider scaling the columns of A to have unit L_1 norm as the most negative strategy proved particularly favourable in this case, and definitely superior to the corresponding scaling using the euclidean norm. Also, for comparison, we give results obtained using a quadratic programming subroutine QUADPR, based on the Cottle-Danzig principal pivoting algorithm, which was supplied by the Madison Academic Computing Center at the University of Wisconsin.

The results for the two data sets are given in Tables 3.1 and 3.2 respectively. We report the average time per problem as (cumulative time)/10 (recorded most unreliably on the computer used, a Univac 1100/42)[†], and the total number of nodes visited. It will be seen that the variants of this algorithm are superior to the quadratic programming algorithm. Also, the most negative heuristic is never too bad, while the stepwise heuristic is favoured for the data drawn from the normal distribution. There is some evidence that the statistical origin of the data is not irrelevant to the choice of a good heuristic. Starting with $\underset{\sim}{\lambda}$ rather than $\underset{\sim}{x}$ is clearly the superior strategy in terms of elapsed time despite the unreliability of the timings (for example, the stepwise and column scaling strategies should have returned approximately the same times in Table 3.1).

The rather dramatic 10-20 fold reduction in time taken for this implementation seems to be partly due to the use of orthogonalisation transformation techniques (as opposed to matrix inversion in the original implementation), and partly due to working with $\underset{\sim}{\lambda}$ rather than $\underset{\sim}{x}$ - so that if, say, $n/2$ elements of $\underset{\sim}{x}$ were non-zero at the optimum, about ¼ of the elements of U in the QU factorisation of A would need to be calculated, as opposed to $^3/4$ if working with $\underset{\sim}{x}$.

[†] Timings in a multiprogramming environment tend to be unreliable because compromises are made between keeping exhaustive records and efficiency. Part of the explanation in this case would appear to stem from the system executive's practice of continuing the internal timing of an interrupted program unless it is actually swapped out of core.

TABLE 3.1

Results for data from normal distribution

Method	Average time (ms)	Number of nodes
Quadratic programming	1187	404
Most negative λ_i	322	210
Stepwise	249	204*
$\| K_j(A) \|_2 = 1$	437	204*
Most negative x_i	547	202

*First feasible solution is optimal for each problem.

TABLE 3.2

Results for data from uniform distribution

Method	Average time (ms)	Number of nodes
Quadratic programming	1340	416
Most negative λ_i	434	196
Stepwise	390	262
$\| K_j(A) \|_2$	385	262
$\| K_j(A) \|_1$	307	168
Most negative x_i	518	240*

*First feasible solution is optimal for each problem.

CHAPTER 4

THE M-ESTIMATOR: STRUCTURE

4.1 INTRODUCTION

4.1.1 Robust Estimation and the Rejection of Outliers

The problem of rejection of outliers may not be quite as old as experimental science, but it has certainly exercised the minds of astronomers, chemists, physicists, etc., for a very long time. The great German astronomer, Bessel, remarked in 1838 that he never rejected an observation merely because of its large residual. Others have not been quite as confident of their equipment and procedures. The first attempt at a rejection criterion based on some sort of probability reasoning seems to have been given by Peirce in 1852. Since then, the topic has become a standard part of least squares theory, and from 1925 onwards has received a great deal of attention from statisticians. An historical review is given by Anscombe (1960), where he also makes the pertinent observation that when a rejection rule is applied, a judgement is not being made on the spuriousness or otherwise of the observation so much as protection is being sought against possible adverse effects - a rejection rule being something akin to an insurance policy.

It is this safeguarding against small deviations from the assumptions that lies at the heart of the search for robust estimators, and any complacency in the use of classical estimators such as the mean square deviation was shattered by Tukey in a rather entertaining article in 1960 with the aid of a simple example. He assumed a randomly mixed batch of "good" and "bad" observations from $N(\mu,\sigma^2)$

and $N(\mu,3\sigma^2)$ distributions respectively. Now Fisher in 1920 had shown that the mean square deviation

$$s_n = \sqrt{\frac{1}{n}\Sigma(x_i-\bar{x})^2}$$

to be 12% more efficient than the mean absolute deviation

$$d_n = \frac{1}{n}\Sigma|x_i-\bar{x}| \ ,$$

where the asymptotic relative efficiency (ARE) is given by

$$\text{ARE} = \lim_{n\to\infty} \frac{\text{var}(s_n)/[E(s_n)]^2}{\text{var}(d_n)/[E(d_n)]^2} \ .$$

The calculations from this example (as corrected by Huber, 1977 (a)) show that although ARE is less than .88 for zero (or 100%) contamination, for as little as 2 bad observations in 1000, ARE is more than 1.00, reaching a maximum of over 2 for 1 bad observation in 20. This is especially disturbing when taken in conjunction with statements such as that of Hampel (1973) "altogether, 5-10% wrong values in a data set seem to be the rule rather than the exception". In that same paper, Hampel gives as the main aim of robust estimation: safeguarding against gross errors; bounding the influence of hidden contaminators; isolating clear outliers; and still being nearly optimal at the strict parameter model.

The search for robust estimators has led to a variety of suggestions, each with its own advocates, (see, eg, Hoffman, 1977; Huber 1977(a), (b); Hampel 1974(a), (b); McKean and Hettmansperger, 1977). Some of these are quite horrendous in the amount of work to be done in identifying more than one or two outliers. Thus to identify k outliers, Andrews (1971) considers projections of the residual vector onto $\binom{n}{k}$ hyperplanes, and Gentleman and Wilk (1975) perform regressions on $\binom{n}{k}$ subsets of data. One wonders whether this last approach is a sufficient improvement over the suggestion of Mickey, Dunn and Clark (1967) who simply do a stepwise regression, using an F-test to determine whether the observation dropped was indeed an outlier. The main approach with all these methods has been to consider the estimator from a statistical point of view, justifying the choice by statistical analysis, sometimes bolstered by Monte-Carlo simulations.

4.1.2 M-estimators

One type of estimator put forward primarily as being distributionally robust (as opposed to model-robust see Hoffman , 1977) is the maximum likelihood or M-estimator.

The classical linear squares estimator is, given a model

$$(4.1) \qquad y_i = \sum_{j=1}^{m} A_{ij} x_j + u_i \quad , \quad i = 1,\ldots,n \ ,$$

where the u_i are independent random errors, to find an m-vector $\underset{\sim}{x}^*$ such that

$$(4.2) \qquad \sum_{i=1}^{n} (r_i^*)^2 = \min,$$

where the residual, r_i , of an observation is given by

$$(4.3) \qquad r_i = \sum_{j=1}^{m} A_{ij} x_j - y_i \ .$$

In an effort to reduce the sensitivity of this estimator to occasional gross errors, Huber (1972) suggested replacing the squared term in (4.2) by a less rapidly increasing function, ρ . Thus we now require $\underset{\sim}{x}^*$ such that

$$(4.4) \qquad \sum_{i=1}^{n} \rho(r_i^*) = \min \ .$$

The favourite choice of ρ by Huber (and others) is

$$(4.5) \qquad \rho(t) = \tfrac{1}{2}t^2 \qquad \text{for } |t| \le c$$
$$= c|t| - \tfrac{1}{2}c^2 \quad \text{for } |t| > c \text{ ,}$$

and it is this function which is the topic of the next two chapters.

This chapter is concerned with the function defined in (4.5), (as applied to residuals in (4.4)), in an effort to understand its underlying structure, as opposed to justifying its statistical virtues. The resulting algorithm will be defined in Chapter 5.

4.2 DEFINITIONS AND PREAMBLE

4.2.1 Definitions and Conventions

A partition P_a is a dividing of the set $N = \{1,2,\ldots,n\}$ into subsets σ_a and $\bar{\sigma}_a$. The function associated with P_a is

$$F_a(\underset{\sim}{x}) = \tfrac{1}{2} \sum_{\sigma_a} r_i(x)^2 + \sum_{\bar{\sigma}_a} \{c|r_i(\underset{\sim}{x})| - \tfrac{1}{2}c^2\} \text{ .}$$

$\underset{\sim}{x}_a$ will refer to the minimiser of $F_a(\underset{\sim}{x})$ and $F_a(\underset{\sim}{x}_a)$ will be called the value of the partition. Residuals of a partition will be measured at its minimum, $\underset{\sim}{x}_a$.

A residual is tight if its absolute value is equal to c .

A partition is tight if at least one of its residuals is tight. "Tightness" will, unless stated otherwise, refer to partitions.

A partition P_a is σ-feasible if $|r_i(\underset{\sim}{x}_a)| \le c$, $i \in \sigma$.

A partition P_a is $\bar{\sigma}$-feasible if $|r_i(\underset{\sim}{x}_a)| > c$, $i \in \bar{\sigma}$.

A partition P_a is feasible if it is σ-feasible and $\bar{\sigma}$-feasible.

The absolute deviation (AD) of a partition P_a is the value of the absolute deviation function $\Sigma|r_i(\underset{\sim}{x})|$, measured at $\underset{\sim}{x}_a$.

The least absolute deviation (LAD) is the global minimum of the AD function.

σ_0 will refer to the set of indexes whose residuals are zero at LAD.

Uniqueness will, unless stated otherwise, refer to the number of feasible partitions, rather than to whether a particular partition has a unique minimum.

θ_i will always be used for the sign of a residual, $\theta_i = \text{sgn}\ r_i(\underset{\sim}{x})$.

Adjacent partitions P_a and P_b satisfy the condition $\sigma_a = \sigma_b \cup \{k\}$.

The examples given in Section 4.3 will always be of the form

$$\begin{array}{c} A^T \\ \underset{\sim}{b}^T \end{array}, \text{ or } \begin{array}{c} A^T \\ \underset{\sim}{b}^T \end{array} \quad c = \ ,$$

thus example 4.1, for instance, has 2 variables and 5 observations. Example 4.1 is given below for convenience, as well as in Section 4.3.1.

<u>Example</u> 4.1

$$\begin{array}{ccccc} 1 & 1 & 4 & 2 & 5 \\ 1 & 4 & 1 & 5 & 3 \\ 1 & 7 & 2 & 12 & 1 \end{array} \qquad c = 2$$

4.2.2 Preamble

In order to get a feel for the structure of the problem - what the function does, what can happen to it, what happens when c is varied - a number of questions and areas of speculation are considered in the next section. In general, the questions posed will be answered in the negative, by a counter example, or will lead to a theorem in the following section. Sometimes, however, the question posed was too simplistic and then a straight "yes/no" will yield to further elaboration.

One area of interest is the relation of the M-estimator to the LAD estimator, where $\rho(t)$ in 4.4 would be

$$(4.6) \qquad \rho(t) = |t| \ .$$

It is clear that, for large enough c , the M-estimator is simply the least squares (LS) estimator. Intuitively, it seems that for small enough c , the M-estimator will be related to the LAD estimator. Several speculations (4.3.3, 4.3.8, 4.3.10, 4.3.11) explore this relationship. The question of non-uniqueness, how it is recognised, under what circumstances it occurs, is considered in 4.3.1, 4.3.6, 4.3.9, 4.3.11. What might be termed the "proper behaviour" of the function gives rise to several speculations (4.3.4, 4.3.5, 4.3.7, 4.3.8, 4.3.10, 4.3.11). Basically this is saying "Can nasty things happen?" "How nasty?". Finally the function value itself is considered in 4.3.2 and 4.3.3.

As Huber (1977,b) observed, once the partitioning $\sigma , \bar{\sigma}$ is known, together with the signs of the residuals in $\bar{\sigma}$, the M-estimator can be calculated very simply. Thus the search for the M-estimator is the search for a feasible partition. The first six questions (4.3.1

to 4.3.6) deal with varying partitions whilst keeping c fixed, and the remaining five (4.3.7 to 4.3.11) deal only with feasible partitions, allowing c to vary.

4.3 SPECULATIONS AND EXAMPLES

4.3.1 On Tightness and Non-uniqueness

This area is possible one of the most interesting in the problem. Given a feasible partition whose minimum is tight, must there be another feasible partition? And, given two feasible partitions, must their respective minima be tight? The answer to this latter question, as will be shown in Section 4.4.1, is "yes" so we now look more closely at the former question. Clearly, we can answer it trivially by taking the partition $\bar{\sigma} = \Phi$ and setting c to the largest sized residual of the least squares solution. It is also true in the more general case $(\bar{\sigma} \neq \Phi)$ that it is possible to have a tight feasible partition which is also the only feasible partition for a particular value of c .

Example 4.1

$$\begin{array}{ccccc} 1 & 1 & 4 & 2 & 5 \\ 1 & 4 & 1 & 5 & 3 \\ 1 & 7 & 2 & 12 & 1 \end{array} \qquad c = 2$$

Here, $\sigma = \{1,2,3,4\}$ has $\underset{\sim}{r}^T = \{.67, 1.33, -2, -2, 2.89\}$, with residuals 3 and 4 being tight. However, as the vectors $\underset{\sim}{a}_1$ and $\underset{\sim}{a}_2$ (corresponding to the non-tight residuals of σ) span the space, this is the only feasible partition (see Theorem 4.1, corollary). This question is pursued a little further in Sections 4.3.9 and 4.3.11.

4.3.2 On the Function Value of Feasible Partitions

As was observed earlier, the search for the M-estimator is the search for a feasible partition. The question then arises as to whether the function value of an infeasible partition can be less than that of a feasible partition, for a given value of c .

Example 4.2

$$\begin{array}{cccccc} 1 & 1 & 1 & 1 & & \\ 0 & 2 & 2.5 & 2.9 & & c = 1 \end{array}$$

Here, the partition $\sigma = \{1\}$ is feasible, and has function value 2.00. However, the partition $\sigma = \{1,4\}$ is infeasible with function value 1.96. This area is explored further in Section 4.4.2 as there are some things which can be shown.

A further question in this area is whether all feasible partition have the same function value. This expected, and hoped-for result, is also shown in Section 4.4.2.

4.3.3 On the Value of the AD Function at Feasible Partitions

The relationship between the M-estimator and the LAD estimator is complex and will be explored further later. Here, we simply observe that the AD can increase from infeasible to feasible partitions.

Example 4.3

$$\begin{array}{ccccccc} 1 & 1 & 1 & 1 & 1 & & \\ -.2 & 2 & 2 & 3.1 & 3.1 & & c = 1 \end{array}$$

Here, the infeasible partition $\bar{\sigma} = \Phi$ has an AD of 4.4 whilst the feasible $\bar{\sigma} = \{1\}$ has an AD of 4.7.

More interestingly, it will be shown (Section 4.4.3) that the AD is the same for all feasible partitions.

4.3.4 On the Signs of Residuals at Change of Partition

As was observed by Huber (1977,b), if the correct partitioning is known, together with the signs of the residuals in $\bar{\sigma}$, the M-estimator can readily be found. However, the signs of residuals can change, even when partitions are adjacent and one of them is feasible.

Example 4.4

$$\begin{array}{cccccccccccccc} 1 & 10 & 1 & 0.9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2.5 & 10 & 3.9 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \end{array} \qquad c = 1$$

For $\sigma = \{2,3\}$, $\underset{\sim}{r}^T = (-1.35, 1.47, -2.75, -4.00, -3.85,\ldots,-3.85)$,

for $\sigma = \{3\}$, $\underset{\sim}{r}^T = (1.3, 28, 0.1, -1.58, -1.2,\ldots,-1.2)$,

so that r_1 $(1 \in \bar{\sigma})$ changes sign between the two partitions, the second of which is feasible. Note that r_3 $(3 \in \sigma)$ also changed sign.

4.3.5 On the Feasibility of Residuals at Change of Partition

Example 4.5 is illustrative of the "anything can happen" property of the problem.

Example 4.5

$$\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 9 & 3 & 3 \end{array} \qquad c = 1$$

For $\sigma = \{1,2,3\}$ $\quad \underset{\sim}{r}^T = (-3,-3,3)$

for $\sigma = \{2,3\}$ $\quad \underset{\sim}{r}^T = (-7,-1,1)$

for $\sigma = \{3\}$ $r = (-5, -2, 1)$ (one solution).

The first change of partition caused a "bad" observation to become "good", without its being involved in the partition change, whilst in the second change of partition we have a "good" observation going "bad". This latter, however, is only possible through the non-uniqueness of the minimum of the partition $\sigma = \{3\}$. The limits on what can happen are given more exactly in Section 4.4.4.

4.3.6 On Non-uniqueness and Connectedness

Thus far, in the examples given when there is more than one feasible partition (examples 4.2, 4.5), the feasible partitions have been connected, i.e. the graph whose nodes are feasible partitions and whose arcs imply adjacency of partitions is connected. This, however, need not be the case (this is relevant also in the proof of Theorem 4.1).

Example 4.6

$$\begin{array}{cccc} 1 & 2 & 2 & 0 \\ 1 & 3 & 0 & 3 \\ 2 & 4 & 3 & 5 \end{array} \qquad c = 1$$

The feasible partitions here are $\sigma = \{4\}$, $\sigma = \{2,3\}$ and $\sigma = \{2,3,4\}$. The last two are connected to each other, but the first is isolated.

Note: In Sections 4.3.7 to 4.3.11 which follow, c will no longer be fixed, and only feasible partitions will be considered.

4.3.7 On σ , as c Varies

The experience of many randomly-generated examples (see Chapter 5) has been that, in general for $c_1 > c_2$, $\bar{\sigma}_{c_2} \supseteq \bar{\sigma}_{c_1}$. However, this need not be the case.

Example 4.7

$$\begin{array}{cccc} 4 & 2 & 8 & 4 \\ 3 & 5 & 5 & 6 \\ 7 & 7 & 12 & 11 \end{array}$$

For $c \geq .62$, $\sigma = \{1,2,3,4\}$.

For $.62 > c \geq .54$, $\sigma = \{1,2,3\}$.

For $.54 > c > .5$, $\sigma = \{1,3\}$,

but for $.5 \geq c \geq .24$, $\sigma = \{1,3,4\}$.

The first example, 4.1, also bears closer scrutiny. For

$2.52 > c > 2$, $\sigma = \{1,2,3\}$.

At $c = 2$, $\sigma = \{1,2,3,4\}$,

whilst for $c < 2$, $\sigma = \{1,2\}$,

illustrating the lack of predictability in the way in which σ varies with c . Lemma 4.1 and Section 4.4.4 will, however, place some limits on this.

4.3.8 On σ_c and σ_0

The preceding section can be extended to include σ_0 , the basis for the LAD. Again, although in general for $c > 0$, $\sigma_0 \subseteq \sigma_c$, this need not be the case.

Example 4.8

$$\begin{array}{cccccc} 1 & 9.5 & 7 & 1 & -5 & 2.466 \\ 1 & 7 & 9 & 3 & 1 & 1.475 \\ 1 & -110 & 90 & 85.6 & -59 & -1 \end{array}$$

For $-1.804 > c > 1.776$, $\sigma_c = \{1,2\}$, the unique feasible partition. However $\sigma_0 = \{3,4\}$, so in this example we have $\sigma_c \cap \sigma_0 = \Phi$.

4.3.9 More on Tightness and Non-uniqueness

In Section 4.3.1, an example was given for which the feasible partition was tight, but unique. In point of fact, for every example for which $n > m$ and is not completely degenerate, as c is decreased there will be values of c at which the feasible partition changes (and in general the feasible partitions will be unique), and at these changes there will be tightness, usually in the old partition ($|\sigma|$ decreasing), but sometimes in the new partition ($|\sigma|$ increasing, as in example 4.7 with $c = 0.5$).

In the following example, the feasible partition is tight for a range of c , but is the only feasible partition in that range.

Example 4.9

$$\begin{array}{cccccc} 1 & 0 & 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 2 & 1/6 & 4 \\ 1 & 1 & 1 & 2 & 8 & 12 \end{array}$$

For $3 \le c < .417$, $\sigma = \{1,2,3,4\}$, and $r_4 = c$.

Note, however, that the non-tight vectors in σ , $\underset{\sim}{a}_1$, $\underset{\sim}{a}_2$ and $\underset{\sim}{a}_3$ span the space, a sufficient condition for uniqueness, though it is not a necessary one - see Section 4.3.11.

4.3.10 On σ_c and σ_0 , for Small c

As was observed earlier, for large enough c , the M-estimator and the L.S. estimator are identical. One would expect that, as $c \to 0$, the M-estimator $\to$ the LAD estimator. It will be shown, in Section 4.4.7, that this is indeed the case, and that, further, $\exists\ \delta > 0$ such that for $c < \delta$, $\sigma_c = \sigma_0$.

4.3.11 On Non-uniqueness of LAD and M-estimators

Clearly, the result alluded to in the preceding section shows that if the LAD is non-unique, then so is the M-estimator for a range of c . The reversed question, whether there can be a non-unique M-estimator, but still a unique LAD, is not as clear. The difficulty lies in the fact that, as will be shown in Section 4.4.1, the non-tight vectors in σ cannot then span the space, and, given that the vectors in σ_0 must span the space, it turns out that by the time σ_c is small enough for there to be another feasible partition, σ_c is getting very close to σ_0 . However, the following example is one in which σ_0 is unique, but for a range of c , σ_c is not.

Example 4.10

$$\begin{array}{ccccc} 1 & 9.5 & 7 & 1 & 3 \\ 1 & 7 & 9 & 3 & 0.5 \\ 1 & -110 & 90 & 85.6 & 58 \end{array}$$

For $c \geq 115$, $\sigma = \{1,2,3,4,5\}$;

for $115 > c \geq 1.869$, $\sigma = \{1,2,3,4\}$;

for $1.869 > c \geq 1.803$, $\sigma = \{1,2,4\}$;

for $1.803 > c > 1.775$, $\sigma = \{1,2\}$, or $\{1,4\}$, or $\{1\}$;

for $c = 1.775$, $\sigma = \{1,2,3\}$, or $\{1,4\}$, or $\{1\}$;

for $1.775 > c > 1.6$, $\sigma = \{1,3\}$, or $\{1,4\}$, or $\{1\}$;

for $1.6 \geq c \geq .384$, $\sigma = \{1,3,4\}$;

for $.384 > c$, $\sigma = \{3,4\}$.

A final point on this particular example. At $c = 1.803$, $\sigma = \{1,2,4\}$, and $\underset{\sim}{r}^T = (-.902, 1.803, -2.007, 1.803, -166.836)$. At $c = 1.6$, $\sigma = \{1,3,4\}$ and $\underset{\sim}{r}^T = (-.8, 3.15, -1.6, 1.6, -166.15)$. In each case the partition is the only feasible one, and in each case the non-tight residuals of σ do not span the space. This again is the exception rather than the rule.

4.4 RESULTS

In this section we prove a number of theorems arising out of the speculations of the previous section, in an attempt to gain greater insight into the structure of the defining function of the M-estimator.

There are a few observations and minor results which will be used freely in the remainder of this section which will be given here rather than repeated each time they are used.

Firstly, any vector $\underset{\sim}{x}$ is feasible for exactly one partition.

As was pointed out before, a set σ defines a function

$$F(\underset{\sim}{x}) = \tfrac{1}{2} \sum_{\sigma} r_i(\underset{\sim}{x})^2 + \sum_{\bar{\sigma}} \{ c|r_i(\underset{\sim}{x})| - \frac{c^2}{2}\} .$$

$F(\underset{\sim}{x})$, being the sum of convex terms is itself convex.

Lemma 4.1

Given $F_a(\underset{\sim}{x})$ with minimiser $\underset{\sim a}{x}$, then provided that $r_i(\underset{\sim a}{x}) \neq 0$, $i \in \sigma$, F_a is strictly convex iff the vectors $\underset{\sim i}{a}$, $i \in \bar{\sigma}$, span the space.

Proof

$$F_a(\underset{\sim}{x}) = \tfrac{1}{2} \sum_{\sigma} r_i(\underset{\sim}{x}) + \sum_{\bar{\sigma}} \{c|r_i(\underset{\sim}{x})| - \frac{c^2}{2}\}$$

$$\nabla F_a(\underset{\sim}{x}) = \sum_{\sigma} r_i(\underset{\sim}{x})\underset{\sim i}{a} + c \sum_{\bar{\sigma}} \theta_i \underset{\sim i}{a} \quad , \quad \text{provided} \quad r_i(\underset{\sim}{x}) \neq 0 \ , \ i \in \bar{\sigma} \ ,$$

$$\text{where} \quad \theta_i = \operatorname{sgn} r_i(\underset{\sim}{x}) \ ,$$

$$= \sum_{\sigma} r_i(\underset{\sim}{x})\underset{\sim i}{a} + \underset{\sim}{b} \ .$$

As $\nabla F_a(\underset{\sim a}{x}) = 0$, $\sum_{\sigma} r_i(\underset{\sim a}{x})\underset{\sim i}{a} + \underset{\sim}{b} = \underset{\sim}{0}$

(i) If also $\nabla F_a(\underset{\sim b}{x}) = \underset{\sim}{0}$

$$\sum_{\sigma} r_i(\underset{\sim a}{x})\underset{\sim i}{a} = \sum_{\sigma} r_i(\underset{\sim b}{x})\underset{\sim i}{a} \ ,$$

and the vectors $\underset{\sim i}{a}$, $i \in \sigma$, do not span the space.

ii) If the vectors $\underset{\sim i}{a}$, $i \in \sigma$, do not span the space we can find a vector $\underset{\sim}{d}$ orthogonal to the $\underset{\sim i}{a}$, $i \in \sigma$, and small enough so that $\operatorname{sgn} r_i(\underset{\sim a}{x}+\underset{\sim}{d}) = \operatorname{sgn} r_i(\underset{\sim a}{x})$, and then $\underset{\sim b}{x} = \underset{\sim a}{x} + \underset{\sim}{d}$ also minimises F_a .

Lemma 4.2

Given adjacent partitions σ_1 and σ_2 with $\sigma_1 = \sigma_2 \cup \{k\}$, then for any vector $\underset{\sim}{x}$ satisfying $\left|r_k(\underset{\sim}{x})\right| = c$, $F_1(\underset{\sim}{x}) = F_2(\underset{\sim}{x})$.

Proof

$$F_1(\underset{\sim}{x}) = \tfrac{1}{2}\sum_{\sigma_1} r_i(\underset{\sim}{x})^2 + \sum_{\bar{\sigma}_1} \{c|r_i(\underset{\sim}{x})| - \frac{c^2}{2}\}$$

$$= \tfrac{1}{2}\sum_{\sigma_2} r_i(\underset{\sim}{x})^2 + \tfrac{1}{2} r_k(\underset{\sim}{x})^2 + \sum_{\bar{\sigma}_1} \{c|r_i(\underset{\sim}{x})| - \frac{c^2}{2}\}$$

$$= \tfrac{1}{2}\sum_{\sigma_2} r_i(\underset{\sim}{x})^2 + c|r_k(\underset{\sim}{x})| - \frac{c^2}{2} + \sum_{\bar{\sigma}_1} c|r_i(\underset{\sim}{x})| - \frac{c^2}{2} \}$$

$$= \tfrac{1}{2}\sum_{\sigma_2} r_i(\underset{\sim}{x})^2 + \sum_{\bar{\sigma}_2} c|r_i(\underset{\sim}{x})| - \frac{c^2}{2} \}$$

$$= F_2(\underset{\sim}{x})$$

Lemma 4.3

Given adjacent partitions σ_1 and σ_2 with $\sigma_1 = \sigma_2 \cup \{k\}$ for which the minimiser $\underset{\sim}{x}^*$ of either has $|r_k(\underset{\sim}{x}^*)| = c$, then $\underset{\sim}{x}^*$ also minimises the other partition.

Proof

$$\nabla F_1(\underset{\sim}{x}^*) = \underset{\sim}{0}$$

$$\Leftrightarrow \sum_{\sigma_1} r_i(\underset{\sim}{x}^*)^2 + c\sum_{\bar{\sigma}_1} \theta_i \underset{\sim}{a}_i = \underset{\sim}{0}$$

$$\Leftrightarrow \sum_{\sigma_2} r_i(\underset{\sim}{x}^*)\underset{\sim}{a}_i + r_k(\underset{\sim}{x}^*)\underset{\sim}{a}_k + c\sum_{\bar{\sigma}_1} \theta_i \underset{\sim}{a}_i = \underset{\sim}{0}$$

$$\Leftrightarrow \sum_{\sigma_2} r_i(\underset{\sim}{x}^*)\underset{\sim}{a}_i + c\sum_{\bar{\sigma}_2} \theta_i \underset{\sim}{a}_i = \underset{\sim}{0}$$

$$\Leftrightarrow \nabla F_2(\underset{\sim}{x}^*) = \underset{\sim}{0}$$

4.4.1 On Tightness and Non-uniqueness

Theorem 4.1 (a)

Given two feasible partitions P_a and P_b such that

$\sigma_b = \sigma_a \cup S$, $S = \{s_1,\ldots,s_r\} \neq \Phi$, then $|r_i(\underset{\sim}{x}_b)| = c$ for all $i \in S$.

Proof

Assume otherwise.

Now as $|r_i(\underset{\sim}{x}_a)| > c$ and $|r_i(\underset{\sim}{x}_b)| \leq c$ for $i \in S$, we can define points $\underset{\sim}{y}_1,\ldots,\underset{\sim}{y}_r$ to satisfy

(i) $\underset{\sim}{y}_i = \theta_i \underset{\sim}{x}_a + (1-\theta_i)\underset{\sim}{x}_b$

(ii) $|r_{s_i}(\underset{\sim}{y}_i)| = c$

(iii) $0 \leq \theta_i \leq \theta_{i+1} \leq 1$

(Thus $\underset{\sim}{y}_i = \alpha_i \underset{\sim}{x}_b + (1-\alpha_i)\underset{\sim}{y}_{i+1}$, $0 \leq \alpha_i \leq 1$, $i = 1,\ldots,r$)

Let $\sigma_i = \sigma_a \cup \{s_{i+1},\ldots,s_r\}$ define $F_i(\underset{\sim}{x})$, $i = 1,\ldots,r-1$, where S has been re-ordered so that s_i corresponds to θ_i .

Now as $\frac{1}{2}t^2 \geq c|t| - \frac{c^2}{2}$ for $c \geq 0$, it follows that

$F_b(\underset{\sim}{x}) \geq F_1(\underset{\sim}{x}) \geq \ldots \geq F_{r-1}(\underset{\sim}{x}) \geq F_a(\underset{\sim}{x})$,

and from the definition of $\underset{\sim}{y}_1,\ldots,\underset{\sim}{y}_r$ and Lemma 4.2 we have

$$F_b(\underset{\sim}{y}_1) = F_1(\underset{\sim}{y}_1)$$

$$F_i(\underset{\sim}{y}_{i+1}) = F_{i+1}(\underset{\sim}{y}_{i+1}) \qquad i = 1,\ldots,r-2$$

$$F_{r-1}(\underset{\sim}{y}_r) = F_a(\underset{\sim}{y}_r)$$

Now,

$$F_1(\underset{\sim}{x}_b) \le F_b(\underset{\sim}{x}_b) \le F_b(\underset{\sim}{y}_1) = F_1(\underset{\sim}{y}_1) \le \alpha_1 F_1(\underset{\sim}{x}_b) + (1-\alpha_1)\ F_1(\underset{\sim}{y}_2)\ ,$$

so

$$F_1(\underset{\sim}{x}_b) \le F_1(\underset{\sim}{y}_2)\ .$$

Again,

$$F_2(\underset{\sim}{x}_b) \le F_1(\underset{\sim}{x}_b) \le F_1(\underset{\sim}{y}_2) = F_2(\underset{\sim}{y}_2) \le \alpha_2 F_2(\underset{\sim}{x}_b) + (1-\alpha_2)\ F_2(\underset{\sim}{y}_3)\ ,$$

so

$$F_2(\underset{\sim}{x}_b) \le F_2(\underset{\sim}{y}_3)\ .$$

$$\vdots$$

$$F_{r-1}(\underset{\sim}{x}_b) \le F_{r-1}(\underset{\sim}{y}_r)\ .$$

Moreover, from our assumption that $|r_i(\underset{\sim}{x}_b)| < c$ for at least some $i \in S$, at least one "$\le$" in this sequence is strictly "$<$", and also $\theta_r > 0$.

We can now write

$$F_{r-1}(\underset{\sim}{x}_b) < F_{r-1}(\underset{\sim}{y}_r)\ .$$

Thus

$$F_a(\underset{\sim}{x}_b) \le F_{r-1}(\underset{\sim}{x}_b) < F_{r-1}(\underset{\sim}{y}_r) = F_a(\underset{\sim}{y}_r) \le (1-\theta_r)F_a(\underset{\sim}{x}_b) + \theta_r F_a(\underset{\sim}{x}_a)\ ,$$

so

$$F_a(\underset{\sim}{x}_b) < F_a(\underset{\sim}{x}_a)\ ,$$

a contradiction, as $\underset{\sim}{x}_a$ is the minimiser of F_a.

This completes the proof.

Note that $\sigma_c = \sigma_a \cup S'$, where $S' \subset S$, is not necessarily a feasible partition. See, eg, example 4.6 where $\sigma = \{4\}$ and $\sigma = \{2,3,4\}$ are feasible partitions, but $\sigma = \{2,4\}$ and $\sigma = \{3,4\}$ are not.

Theorem 4.1 (b)

Given two feasible partitions P_a and P_b ,

let $\sigma = \sigma_a \cap \sigma_b$.

Then $|r_i(\underset{\sim}{x}_a)| = c$ for $i \in \sigma_a \cap \bar{\sigma} = \bar{\sigma}_a \cap \sigma_b$

$|r_i(\underset{\sim}{x}_b)| = c$ for $i \in \sigma_b \cap \bar{\sigma} = \bar{\sigma}_b \cap \sigma_a$

Proof

Let $\sigma_c = \sigma_a \cap \sigma_b$.

Let $S_a = \sigma_a \cap \bar{\sigma}_b = \sigma_a \cap \bar{\sigma}_c$.

Let $S_b = \sigma_b \cap \bar{\sigma}_a = \sigma_b \cap \bar{\sigma}_c$.

Then $\sigma_a = \sigma_c \cup S_a$, and $\sigma_b = \sigma_c \cup S_b$.

Let $\underset{\sim}{y}_c = \theta_c \underset{\sim}{x}_a + (1-\theta_c)\underset{\sim}{x}_b$, $0 \le \theta_c \le 1$,

such that $|r_i(\underset{\sim}{y}_c)| \le c$, $i \in \sigma_c$.

Moreover, as $\underset{\sim}{y}_c$ cannot be feasible for σ_a or σ_b (being feasible for σ_c) , $\theta_c \ne 0$ and $\theta_c \ne 1$. Then from the same argument as before,

$$F_c(\underset{\sim}{x}_b) \le F_c(\underset{\sim}{y}_c) \text{ , and } F_c(\underset{\sim}{x}_a) \le F_c(\underset{\sim}{y}_c) .$$

Hence $\theta_c F_c(\underset{\sim}{x}_a) + (1-\theta_c)F_c(\underset{\sim}{x}_b) \leq F_c(\underset{\sim}{y}_c)$.

Now assume that for at least some $i \in \sigma_a \cap \bar{\sigma}_c$, $|r_i(\underset{\sim}{x}_a)| < c$. Then, as before, $F_c(\underset{\sim}{x}_a) < F_c(\underset{\sim}{y}_c)$, and, as $\theta_c \neq 0$, we have

$$\theta_c F_c(\underset{\sim}{x}_a) + (1-\theta_c)F_c(\underset{\sim}{x}_b) < F_c(\underset{\sim}{y}_c) .$$

But, from convexity,

$$F_c(\underset{\sim}{y}_c) \leq \theta_c F_c(\underset{\sim}{x}_a) + (1-\theta_c)F_c(\underset{\sim}{x}_b) ,$$

a contradiction.

Hence $|r_i(\underset{\sim}{x}_a)| = c$ for all $i \in \sigma_a \cap \bar{\sigma}_c$

Similarly $|r_i(\underset{\sim}{x}_b)| = c$ for all $i \in \sigma \cap \bar{\sigma}_c$.

It will be observed that Theorem 4.1 (a) is merely a special case of Theorem 4.1 (b), with $\sigma_a \cap \bar{\sigma}_b = \Phi$. The splitting of the theorem into two was done in the interests of clarity, in order to treat the simpler case first.

Corollary

A necessary condition for non-uniqueness of a partition P_a is that the non-tight vectors of σ_a do not span the space.

Proof

Let P_a and P_b be non-unique partitions.

Define $\sigma_c = \sigma_a \cap \sigma_b$.

Then, from Theorem 4.1 (b) and Lemma 4.3, $\underset{\sim}{x}_a$ minimises $F_c(\underset{\sim}{x})$, and $\underset{\sim}{x}_b$ minimises $F_c(\underset{\sim}{x})$.

Moreover as $\underset{\sim}{x}_a$ is feasible for P_a and $\underset{\sim}{x}_b$ for P_b ,

$\underset{\sim}{x}_a \neq \underset{\sim}{x}_b$, thus from Lemma 4.1, the vectors of $\sigma_c = \sigma_a \cap \sigma_b$ do not span the space. And as all vectors of $\sigma_a \cap \bar{\sigma}_b$ are tight, σ_c must include all non-tight vectors of σ_a .

We now have two necessary conditions for non-uniqueness, namely tightness, and that the non-tight vectors of σ do not span the space. These two conditions, however, are not sufficient to ensure non-uniqueness (except, of course, when there is only one tight residual), see example 4.10.

4.4.2 On Function Values of Feasible Partitions

Theorem 4.2 (a)

If P_a is a feasible partition, and P_b is σ-feasible, then $F_a(\underset{\sim}{x}_a) \geq F_b(\underset{\sim}{x}_b)$.

Proof

Define P_c by $\sigma_c = \sigma_a \cap \sigma_b$.

Then by the same argument as was used in Theorem 4.1, as P_b is σ-feasible and P_a is $\bar{\sigma}$-feasible,

$$F_b(\underset{\sim}{x}_b) \leq F_c(\underset{\sim}{x}_a)$$

But as $\sigma_c \subset \sigma_a$,

$$F_c(\underset{\sim}{x}_a) \leq F_a(\underset{\sim}{x}_a) ,$$

$$\therefore \quad F_b(\underset{\sim}{x}_b) \leq F_a(\underset{\sim}{x}_a) .$$

Theorem 4.2 (b)

If P_a is feasible and P_b is $\bar{\sigma}$-feasible, then

$$F_a(\underset{\sim}{x}_a) \leq F_b(\underset{\sim}{x}_b)$$

Proof

Similar to above.

From the above theorems, we see that the feasible partition has a function value smallest of all the $\bar{\sigma}$-feasible partitions, and largest of all σ-feasible ones.

Corollary

The function values of all feasible partitions are equal.

Note: This property can also be inferred directly from the convexity of $F(\underset{\sim}{x})$.

4.4.3 On Signs and Values of Residuals in Non-unique Partitions

Lemma 4.4

Given a partition with non-unique minimisers $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ of F , then

(i) $r_i(\underset{\sim}{x}_1) = r_i(\underset{\sim}{x}_2) \quad i \in \sigma$

(ii) $\operatorname{sgn} r_i(\underset{\sim}{x}_1) = \operatorname{sgn} r_i(\underset{\sim}{x}_2) \ , \quad i \in \bar{\sigma}$

Proof

Let $\underset{\sim}{x}_3 = \frac{1}{2}\underset{\sim}{x}_1 + \frac{1}{2}\underset{\sim}{x}_2$.

Then, from the convexity of F ,

$$F(\underset{\sim}{x}_3) \leq \tfrac{1}{2}F(\underset{\sim}{x}) + \tfrac{1}{2}F(\underset{\sim}{x}_2) = F(\underset{\sim}{x}_1)$$

so $\quad F(\underset{\sim}{x}_3) = \tfrac{1}{2}F(\underset{\sim}{x}_1) + \tfrac{1}{2}F(\underset{\sim}{x}_2)$

i.e.
$$\tfrac{1}{2}\sum_{\sigma} r_i\left(\frac{\underset{\sim}{x}_1+\underset{\sim}{x}_2}{2}\right)^2 + \sum_{\bar{\sigma}}\left\{c\left|r_i\left(\frac{\underset{\sim}{x}_1+\underset{\sim}{x}_2}{2}\right)\right| - \frac{c^2}{2}\right\}$$

$$= \tfrac{1}{4}\sum_{\sigma}\{r_i(\underset{\sim}{x}_1)^2 + r_i(\underset{\sim}{x}_2)^2\} + \tfrac{1}{2}\sum_{\bar{\sigma}}\{c|r_i(\underset{\sim}{x}_1)| + c|r_i(\underset{\sim}{x}_2)| - c^2\}\ .$$

Now
$$r_i\left(\frac{\underset{\sim}{x}_1+\underset{\sim}{x}_2}{2}\right) = \tfrac{1}{2}r_i(\underset{\sim}{x}_1) + \tfrac{1}{2}r_i(\underset{\sim}{x}_2)\ ,$$

so
$$\frac{1}{8}\sum_{\sigma}\{r_i(\underset{\sim}{x}_1) + r_i(\underset{\sim}{x}_2)\}^2 + \frac{c}{2}\sum_{\bar{\sigma}}|r_i(\underset{\sim}{x}_1) + r_i(\underset{\sim}{x}_2)|$$

$$= \tfrac{1}{4}\sum_{\sigma}\{r_i(\underset{\sim}{x}_1)^2 + r_i(\underset{\sim}{x}_2)\}^2 + \frac{c}{2}\sum_{\bar{\sigma}}\{|r_i(\underset{\sim}{x}_1)| + |r_i(\underset{\sim}{x}_2)|\}\ .$$

But $\quad \tfrac{1}{2}(p+q)^2 \le p^2 + q^2$, and $|p+q| \le |p| + |q|$,

so equality can only occur when all corresponding elements are equal,

i.e. $\quad r_i(\underset{\sim}{x}_1) = r_i(\underset{\sim}{x}_2) \quad i \in \sigma$

$\qquad \operatorname{sgn} r_i(\underset{\sim}{x}_1) = \operatorname{sgn} r_i(\underset{\sim}{x}_2) \quad i \in \bar{\sigma}$

Theorem 4.3

Given non-unique feasible partitions P_a and P_b , let $\sigma = \sigma_a \cap \sigma_b$. Then

(i) $\quad r_i(\underset{\sim}{x}_a) = r_i(\underset{\sim}{x}_b)$, $i \in \sigma$

(ii) $\quad \operatorname{sgn} r_i(\underset{\sim}{x}_a) = \operatorname{sgn} r_i(\underset{\sim}{x}_b)$, $i \in \bar{\sigma}$.

Proof

Define P_c by $\sigma_c = \sigma_a \cap \sigma_b$.

Then, from Lemma 4.4, $\underset{\sim}{x}_a$ and $\underset{\sim}{x}_b$ are non-unique minima of F_c . The proof follows.

Corollary

The AD function is the same at all feasible solutions.

Proof

Let $\underset{\sim}{x}_a$ and $\underset{\sim}{x}_b$ be non-unique solutions.

Then, from Theorem 4.2 corollary,

$$\sum_{\sigma_a} r_i(\underset{\sim}{x}_a)^2 + \sum_{\bar{\sigma}_a}\{c|r_i(\underset{\sim}{x}_a)| - \frac{c^2}{2}\} = \sum_{\sigma_b} r_i(\underset{\sim}{x}_b)^2 + \sum_{\bar{\sigma}_b}\{c|r_i(\underset{\sim}{x}_b)| - \frac{c^2}{2}\}$$

Let $\sigma_c = \sigma_a \cap \sigma_b$.

Then, from Theorem 4.1 and Lemma 4.2

$$\sum_{\sigma_c} r_i(\underset{\sim}{x}_a)^2 + \sum_{\bar{\sigma}_c} \{c|r_i(\underset{\sim}{x}_a)| - \frac{c^2}{2}\} = \sum_{\sigma_c} r_i(\underset{\sim}{x}_b)^2 + \sum_{\bar{\sigma}_c} \{c|r_i(\underset{\sim}{x}_b)| - \frac{c^2}{2}\} .$$

But, from Theorem 4.3, $r_i(\underset{\sim}{x}_a) = r_i(\underset{\sim}{x}_b)$, $i \in \sigma_c$,

$$\therefore \quad \sum_{\bar{\sigma}_c} \{c|r_i(\underset{\sim}{x}_a)| - \frac{c^2}{2}\} = \sum_{\bar{\sigma}_c} \{c|r_i(\underset{\sim}{x}_b)| - \frac{c^2}{2}\}$$

$$\therefore \quad \sum_{\bar{\sigma}_c} |r_i(\underset{\sim}{x}_a)| = \sum_{\bar{\sigma}_c} |r_i(\underset{\sim}{x}_b)| .$$

Also, as $r_i(\underset{\sim}{x}_a) = r_i(\underset{\sim}{x}_b)$, $i \in \sigma_c$,

$$\sum_{\sigma_c} |r_i(\underset{\sim}{x}_a)| = \sum_{\sigma_c} |r_i(\underset{\sim}{x}_b)| ,$$

hence

$$\sum |r_i(\underset{\sim}{x}_a)| = \sum |r_i(\underset{\sim}{x}_b)| \ .$$

4.4.4 On the Size of Residuals at Change of Partition

Theorem 4.4

Let P_a and P_b be adjacent partitions, with unique minima $\underset{\sim}{x}_a$ and $\underset{\sim}{x}_b$, such that $\sigma_a = \sigma_b \cup \{k\}$, then

(i) $|r_k(\underset{\sim}{x}_a)| > c \Rightarrow |r_k(\underset{\sim}{x}_b)| > c$

(ii) $|r_k(\underset{\sim}{x}_b)| \le c \Rightarrow |r_k(\underset{\sim}{x}_a)| \le c$

(iii) $|r_k(\underset{\sim}{x}_a)| \le c \Rightarrow |r_k(\underset{\sim}{x}_b)| \le c$

(iv) $|r_k(\underset{\sim}{x}_b)| > c \Rightarrow |r_k(\underset{\sim}{x}_a)| > c$.

Proof

For (i) and (ii), assume $|r_k(\underset{\sim}{x}_a)| > c$ and $|r_k(\underset{\sim}{x}_b)| \le c$.

Define $\underset{\sim}{x}_c = \theta \underset{\sim}{x}_a + (1-\theta)\underset{\sim}{x}_b$, $0 \le \theta < 1$, such that $|r_k(\underset{\sim}{x}_c)| = c$.

Then

$$F_b(\underset{\sim}{x}_a) \le F_a(\underset{\sim}{x}_a) \le F_a(\underset{\sim}{x}_c) = F_b(\underset{\sim}{x}_c) \le \theta F_b(\underset{\sim}{x}_a) + (1-\theta)F_b(\underset{\sim}{x}_b)$$

$\therefore \quad F_b(\underset{\sim}{x}_a) \le F_b(\underset{\sim}{x}_b)$, as $\theta < 1$,

a contradiction.

For (iii) and (iv), assume $|r_k(\underset{\sim}{x}_a)| \le c$ and $|r_k(\underset{\sim}{x}_b)| > c$.

Define $\underset{\sim}{x}_c = \theta \underset{\sim}{x}_a + (1-\theta)\underset{\sim}{x}_b$, $0 < \theta \le 1$, such that

$|r_k(\underset{\sim}{x}_c)| = c$.

Then $F_b(\underset{\sim}{x}_a) \leq F_a(\underset{\sim}{x}_a) \leq F_a(\underset{\sim}{x}_c) = F_b(\underset{\sim}{x}_c) \leq \theta F_b(\underset{\sim}{x}_a) + (1-\theta)F_b(\underset{\sim}{x}_b)$,

so $F_b(\underset{\sim}{x}_a) \leq F_b(\underset{\sim}{x}_b)$, unless $\theta = 1$, in which case $\underset{\sim}{x}_b$ also minimises F_a . In either case there is a contradiction.

Note that Example 4.5 is not a counter example of this theorem as that in that example one of the partitions did not have a unique minimum.

4.4.5 On the Composition of σ and $\bar{\sigma}$

Theorem 4.5

If P_a is a feasible partition, and it is known that $F_b(\underset{\sim}{x}_b) < F_a(\underset{\sim}{x}_a)$, and $F_c(\underset{\sim}{x}_c) > F_a(\underset{\sim}{x}_a)$, then

(i) $\sigma_a \cap \bar{\sigma}_{b\leq} \neq \Phi$, where $\bar{\sigma}_{b\leq} = \{i | i \in \bar{\sigma}_b, |r_i(\underset{\sim}{x}_b)| \leq c\}$

(ii) $\bar{\sigma}_a \cap \sigma_{c>} \neq \Phi$, where $\sigma_{c>} = \{i | i \in \sigma_c, |r_i(\underset{\sim}{x}_c)| > c\}$.

Proof

(i) Assume otherwise, i.e. for all $i \in \sigma_a \cap \bar{\sigma}_b$, $|r_i(\underset{\sim}{x}_b)| > c$. Then, by an argument, similar to that used in Theorem 4.1, we have $F_a(\underset{\sim}{x}_a) \leq F_b(\underset{\sim}{x}_b)$, a contradiction.

(ii) Similar to (i) .

Note: P_b may be, but does not have to be, σ-feasible,

P_c may be, but does not have to be, $\bar{\sigma}$-feasible.

4.4.6 On the Size of σ

Theorem 4.6

Providing there are m linear independent (row) vectors in A, then there exists a feasible partition for which there are at least m linearly independent vectors in σ.

Proof

Assume we have a feasible partition with fewer than m linearly independent vectors in σ. Then, from Lemma 4.1, there will not be a unique minimum for that partition, and the minimum will be portion of a hyperplane bounded by hyperplanes defined by $|r_i(\underset{\sim}{x})| = c$, $i \in \bar{\sigma}$. At any point of intersection, we have a new partition which, from Theorem 4.3, is feasible, and with one more linearly independent vector in σ. As long as the minimum is not unique, i.e. there are fewer than m linearly independent vectors in σ, this process can be repeated.

4.4.7 On the Connection with LAD for Small c

Theorem 4.7

For small enough, but positive, c, $\sigma_c = \sigma_0$.

Proof

Let $\sigma = \sigma_0$,

then $$F(\underset{\sim}{x}) = \tfrac{1}{2}\sum_{\sigma} r_i(\underset{\sim}{x})^2 + \sum_{\bar{\sigma}} \{c|r_i(\underset{\sim}{x})| - \frac{c^2}{2}\} .$$

Let $\underset{\sim}{x}^*$ minimise F.

Hence

$$\frac{dF}{dx}\,(\underset{\sim}{x}^*) = \sum_{\sigma} r_i(\underset{\sim}{x}^*)\underset{\sim}{a}_i + \sum_{\bar{\sigma}} c\theta_i\underset{\sim}{a}_i = \underset{\sim}{0} \cdot$$

Thus,

$$\sum_{\sigma} \frac{dr_i(\underset{\sim}{x}^*)}{dc}\,\underset{\sim}{a}_i + \sum_{\bar{\sigma}} \theta_i\underset{\sim}{a}_i = \underset{\sim}{0} \ .$$

Now, from the characterization of the LAD optimum, (see, eg, Watson, 1980), we have $\exists\,\underset{\sim}{\lambda}$ such that

$$\sum_{\sigma} \lambda_i\underset{\sim}{a}_i + \sum_{\bar{\sigma}} \theta_i\underset{\sim}{a}_i = \underset{\sim}{0} \ , \text{ and } |\underset{\sim}{\lambda}| \le \underset{\sim}{1}$$

But as the $\underset{\sim}{a}_i$, $i \in \sigma$, span the space,

$$\frac{dr_i(\underset{\sim}{x}^*)}{dx} = \lambda_i$$

i.e. $\left|\dfrac{dr_i(\underset{\sim}{x}^*)}{dx}\right| \le 1$.

and, as $r_i(\underset{\sim}{x}^*) = 0$ when $c = 0$ for $i \in \sigma$,

$$|r_i(\underset{\sim}{x}^*)| \le c \ .$$

Thus the residuals within σ stay feasible as c is varied.

Now $r_i(\underset{\sim}{x}^*) \neq 0$, $c = 0$, $i \in \sigma$. Thus for $c < \delta$ $|r_i(\underset{\sim}{x}^*)| > c$,

where $$\delta = \min_{\bar{\sigma}} \left\{ \frac{|r_i(\underset{\sim}{x}_0)|}{1\text{-sgn } r_i(\underset{\sim}{x}_0)\frac{dr_i}{dc}} \;\middle|\; \text{sgn } r_i(\underset{\sim}{x}_0)\,\frac{dr_i}{dc} < 1 \right\}$$

and $\delta > 0$ as $\dfrac{dr_i}{dc}$ is finite.

Thus for the range of c , $0 \le c < \delta$, σ_0 is a feasible partition.

Corollary

If the LAD is non-unique, there is a non-unique M-estimator for a range of c .

4.5 SUMMARY

We have seen that the defining function of the M-estimator is not a simple one. Odd things can happen (eg, example 4.4), which can make the finding of a feasible partition difficult. We do, however, know some limits on the behaviour at change of partition, and where a partition is tight, the function is well-behaved.

We have been able to establish the relationship with the LAD estimator, its extent and its limits.

We have been able to establish necessary conditions for recognising non-uniqueness, and although they are not quite sufficient, we can recognise those times that they are.

Examples such as 4.7 where $\bar{\sigma} = \{4\}$ for one range of values of c , and $\{2\}$ for another pose the question "which is the outlier?". A final comment from Andrews (1971) is relevant here "Such observations should not be rejected, but rather receive special attention. To ignore them would appreciably limit the information to be gained from the current and subsequent experiments".

CHAPTER 5

AN ALGORITHM FOR THE M-ESTIMATOR

5.1 VARIOUS APPROACHES

Along with the rapid development and great interest in the theory of robust estimators, there has naturally arisen a corresponding interest in algorithms to compute, in particular, the M-estimator. Several approaches have been suggested, and some of these have been developed and extensively tested. Of these, approaches which attempt to find a specific number of outliers by considering in some manner all possible subsets (eg Andrews, 1971, Gentleman and Wilk, 1975) do not appear promising, both because of the amount of work to be done and because of possible ambiguity in the answer. Example 4.7, when in the case of just a single outlier, the identity of the outlier can vary for different ranges of c, illustrates this.

A number of iterative methods have been developed, and the most popular of these are summarised below. In some of these a scaling factor is estimated by the algorithm at each iteration (eg Huber, 1973, Huber and Dutter, 1974), and some just estimate it once before computation begins (eg Beaton and Tukey, 1974, Holland and Welsch, 1977). For simplicity, scaling will be omitted below.

We are concerned with the problem

(5.1) $\min \Sigma \rho(r_i)$,

or, equivalently, if ρ is convex and differentiable

(5.2) $$\Sigma\, \psi(r_i) = 0$$

where $\psi = \rho'$ and r_i is the i^{th} residual

(5.3) $$r_i = \sum_{j=1}^{m} A_{ij} x_j - y_i \, .$$

Three iterative schemes have been suggested for solving (5.2).

(5.4) $$\underset{\sim}{x}^{i+1} = \underset{\sim}{x}^{i} + (A^T < \psi'(\underset{\sim}{r}^i) > A)^{-1} \; A^T \psi(\underset{\sim}{r}^i)$$

(5.5) $$\underset{\sim}{x}^{i+1} = \underset{\sim}{x}^{i} + (A^T A)^{-1} \; A^T \psi(\underset{\sim}{r}^i)$$

(5.6) $$\underset{\sim}{x}^{i+1} = \underset{\sim}{x}^{i} + (A^T < w(\underset{\sim}{r}^i) > A)^{-1} \; A \; < w(\underset{\sim}{r}^i) > \underset{\sim}{r}^i$$

where $< \underset{\sim}{a} >$ denotes a diagonal matrix with diagonal elements $< >_{ii} = a_i$, and w is a weighting function.

(5.4) is simply the Newton method and has been applied by Huber and Dutter (1974). According to Holland and Welsch (1977) it is the fastest, but difficult to implement as it requires ψ' , and $A^T < \psi' > A$ may be negative definite. (5.5) is Huber and Dutter's (1974) method, and they describe it as the usual least squares method with the residuals being "Winsorised". It has the desirable property that the generalised inverse $(A^T A)^{-1} A^T$ need be calculated only once, but Holland and Welsch report it as being the slowest of the three methods and not being easy to use with existing least-squares packages. (5.6) is the iteratively reweighted least squares method due to Beaton and Tukey (1974) (although Schlossmacher, 1973, does use an iteratively reweighted least squares method for the least absolute deviation estimator). Detailed examinations of the method are given by Holland and Welsch (1977) and Byrd and Pyne (1979).

For the M-estimator,

$$(5.7) \qquad \min \Sigma\, \rho(r_i)$$

where $\rho(t) = \tfrac{1}{2}t^2$ for $|t| \le c$

$$= c|t| - \tfrac{1}{2}c^2 \quad \text{for} \quad |t| > c\ ,$$

Huber (Huber, 1973, Dutter, 1977) proposed a method based on the idea that if the partitioning $\sigma = \{i \mid -c \le r_i \le c\}$, $\bar{\sigma}_+ = \{i \mid r_i > c\}$, $\bar{\sigma}_- = \{i \mid r_i < -c\}$ is known at the optimum, the problem is simple. The algorithm starts at an initial estimate, determines the partitions σ , $\bar{\sigma}_+$, $\bar{\sigma}_-$ and solves (5.7) on the assumption that this partitioning is correct. This process is repeated until the partitioning does not change between successive iterations. Huber (1973) reports that this method is fast, but can run into singularity problems.

The algorithm presented below uses the result that the M-estimator, i.e. the optimiser of (5.7), is a continuous piecewise linear function of c . This means that once the correct partitioning has been found for a particular value of c , that partitioning is correct for a range of c . Also, residuals becoming infeasible when the range is exceeded indicate the correct partitioning for the next range. Once this is done, the M-estimator for the next range can be found with little additional work. The initial step is to find the least squares solution, corresponding to taking c arbitrarily large, and for which $\bar{\sigma}_+ = \bar{\sigma}_- = \Phi$. The algorithm then proceeds by decreasing c in steps until the correct value of c is reached. Thus the algorithm finds the outliers (as defined by the M-estimator) for all values of c greater than any desired value, a useful feature given the possible ambiguity of "one outlier". The algorithm falls into the category of continuation algorithms (Ortega and Rheinboldt, 1970), with c being the continuation parameter.

5.2 THE ALGORITHM

5.2.1 Piecewise Linearity

Theorem 5.1

The M-estimator is a continuous piecewise linear function of c .

Proof

If we re-write (5.7) as

$$\text{(5.8)} \qquad \text{minimise } F(c,\underset{\sim}{x}) = \tfrac{1}{2}\sum_{\sigma} r_i^2 + \sum_{\bar{\sigma}} \{c|r_i| - \tfrac{1}{2}c^2\} ,$$

where $\sigma = \{i|\ |r_i(\underset{\sim}{x}^*)| \le c\}$, then, for any c , the condition for a minimum gives

$$\text{(5.9)} \qquad \underset{\sim}{0} = \sum_{\sigma} r_i(\underset{\sim}{x}^*)\underset{\sim}{a}_i + \sum_{\bar{\sigma}} c\theta_i \underset{\sim}{a}_i ,$$

where $\theta_i = \text{sgn } r_i(\underset{\sim}{x}^*)$.

(For the remainder of this chapter, we will only be discussing optimal points, so $\underset{\sim}{x}$ will be used for $\underset{\sim}{x}^*$ and r_i for $r_i(\underset{\sim}{x}^*)$).

Differentiating (5.9) with respect to c gives

$$\text{(5.10)} \qquad \underset{\sim}{0} = \sum_{\sigma} \underset{\sim}{a}_i \frac{dr_i}{dc} + \sum_{\bar{\sigma}} \theta_i \underset{\sim}{a}_i ,$$

or

$$\underset{\sim}{0} = \sum_{\sigma} \underset{\sim}{a}_i \underset{\sim}{a}_i^T \frac{d\underset{\sim}{x}}{dc} + \sum_{\bar{\sigma}} \theta_i \underset{\sim}{a}_i$$

$$(5.11) \qquad = B^T B \frac{d\underset{\sim}{x}}{dc} + \sum_{\bar{\sigma}} \theta_i \underset{\sim}{a}_i \, ,$$

where B is the submatrix of A defined by σ , i.e.

$\underset{\sim}{b}_j = \underset{\sim}{a}_i$, $i \in \sigma$, $j = 1,\ldots,|\sigma|$, where $\underset{\sim}{b}_j$ and $\underset{\sim}{a}_i$ correspond to rows of B and A respectively.

Differentiating (5.11) we see that $\frac{d^2\underset{\sim}{x}}{dc^2} = \underset{\sim}{0}$, so $\frac{d\underset{\sim}{x}}{dc}$ and $\frac{dr_i}{dc}$ are piecewise constant, and $\underset{\sim}{x}$ and r_i are piecewise linear in c . Moreover, at the end of a range, where extending it further would make $|r_k| > c$ for some $k \in \sigma$ or $|r_k| \leq c$ for some $k \in \bar{\sigma}$, $|r_k| = c$ and so, by Lemma 4.3, the optimiser of (5.8) also solves (5.8) for $\sigma' = \sigma \pm \{k\}$, which is the new partition. Hence $\underset{\sim}{x}$ is continuous.

5.2.2 Updating at Change of Partition

In the previous section, we showed that $\underset{\sim}{x}$ was piecewise linear in c . When the range changes, there will be a new partition. We now show how to carry out the changes so involved efficiently.

If we can find a positive definite matrix P such that $B^T B = PP^T$, then (5.11) becomes

$$(5.12) \qquad P^T \frac{d\underset{\sim}{x}}{dc} = -\sum_{\bar{\sigma}} \theta_i P^{-1} \underset{\sim}{a}_i$$

$$= -\sum_{\bar{\sigma}} \theta_i \underset{\sim}{w}_i$$

where $\underset{\sim}{w}_i = P^{-1}\underset{\sim}{a}_i$. Then

$$(5.13) \qquad \frac{dr_j}{dc} = \underset{\sim}{a}_j^T \frac{d\underset{\sim}{x}}{dc} = -\underset{\sim}{w}_j^T \sum_{\bar{\sigma}} \theta_i \underset{\sim}{w}_i \ .$$

At a change of partition $\sigma \rightarrow \sigma \pm \{k\}$,

$$(5.14) \qquad PP^T \rightarrow PP^T \pm \underset{\sim}{a}_k \underset{\sim}{a}_k^T = P\{I \pm \underset{\sim}{w}_k \underset{\sim}{w}_k^T\} P^T = P'P'^T \ .$$

Now for any orthogonal transformation $QQ^T = I$,

$$(5.15) \qquad I \pm \underset{\sim}{w}_k \underset{\sim}{w}_k^T = QQ^T \pm QQ^T \underset{\sim}{w}_k \underset{\sim}{w}_k^T QQ^T$$

$$= Q\,\{I \pm Q^T \underset{\sim}{w}_k \underset{\sim}{w}_k^T Q\}\,Q^T \ .$$

If we select Q such that

$$(5.16) \qquad Q^T \underset{\sim}{w}_k = \|\underset{\sim}{w}_k\|\ \underset{\sim}{e}_\alpha \ , \qquad 1 \leq \alpha \leq m$$

(5.15) becomes

$$(5.17) \qquad I \pm \underset{\sim}{w}_k \underset{\sim}{w}_k^T = Q\,\{I \pm \|\underset{\sim}{w}_k\|^2 \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T\}\,Q^T$$

$$= Q\,D^{\frac{1}{2}}\,(D^{\frac{1}{2}})^T Q^T$$

so, from (5.14),

$$(5.18) \qquad P' = P\,Q\,D^{\frac{1}{2}} \ ,$$

and from (5.12)

$$(5.19) \qquad \underset{\sim}{w}_j' = P'^{-1}\underset{\sim}{a}_j = D^{-\frac{1}{2}} Q^T \underset{\sim}{w}_j \ .$$

The matrix Q can be calculated from the Householder transformation (see, eg, Wilkinson and Reinsch, 1971).

$$(5.20) \qquad (I - 2\,\hat{\underset{\sim}{q}}\,\hat{\underset{\sim}{q}}^T)\underset{\sim}{w}_k = \theta_k\,\|\underset{\sim}{w}_k\|\ \underset{\sim}{e}_\alpha \ ,$$

$$(5.21) \qquad (2\underset{\sim}{\hat{q}}^T \underset{\sim}{w}_k)\underset{\sim}{\hat{q}} = \underset{\sim}{w}_k - \theta_k \| \underset{\sim}{w}_k \| \underset{\sim}{e}_\alpha \; ,$$

$$(5.22) \qquad 2\underset{\sim}{\hat{q}}^T \underset{\sim}{w}_k = \{ 2(\underset{\sim}{w}_k^T \underset{\sim}{w}_k - \theta_k \| \underset{\sim}{w}_k \| (\underset{\sim}{w}_k)_\alpha \}^{\frac{1}{2}} \; .$$

After the $\underset{\sim}{w}_j$ have been updated in this manner, the $\frac{dr_j}{dc}$ can be updated rather efficiently in the following way.

From (5.13) and (5.19),

$$\frac{dr_j'}{dc} = -\underset{\sim}{w}_j'^T \sum_{\bar{\sigma}'} \theta_i \underset{\sim}{w}_i'$$

$$(5.23) \qquad = -\underset{\sim}{w}_j^T Q D^{-\frac{1}{2}} \sum_{\bar{\sigma}} \theta_i D^{-\frac{1}{2}} Q^T \underset{\sim}{w}_j \pm \underset{\sim}{w}_j'^T \underset{\sim}{w}_k' \; .$$

Now, from (5.17)

$$D^{-1} = \{ I \pm \| \underset{\sim}{w}_k \|^2 \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T \}^{-1}$$

$$(5.24) \qquad = \left\{ I \mp \frac{\| \underset{\sim}{w}_k \|^2 \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T}{1 \pm \| \underset{\sim}{w}_k \|^2} \right\} \; ,$$

and from (5.16) and (5.19) $\underset{\sim}{w}_k'$ is parallel to $\underset{\sim}{e}_\alpha$, so

$$(5.25) \qquad \underset{\sim}{w}_j'^T \underset{\sim}{w}_k' = (\underset{\sim}{w}_j')_\alpha \, (\underset{\sim}{w}_k')_\alpha \; .$$

Applying (5.24) and (5.25) to (5.23), we have

$$\frac{dr_j'}{dc} = -\sum_{\bar{\sigma}} \theta_i \underset{\sim}{w}_j^T \underset{\sim}{w}_i \pm \frac{\| \underset{\sim}{w}_k \|^2}{1 \pm \| \underset{\sim}{w}_k \|^2} \sum_{\bar{\sigma}} \theta_i \underset{\sim}{w}_j^T Q \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T Q^T \underset{\sim}{w}_i \pm (\underset{\sim}{w}_j')_\alpha (\underset{\sim}{w}_k')_\alpha$$

$$= \frac{dr_j}{dc} \pm \frac{\| \underset{\sim}{w}_k \|^2}{1 \pm \| \underset{\sim}{w}_k \|^2} \sum_{\bar{\sigma}} \theta_i (Q^T \underset{\sim}{w}_j)_\alpha (Q^T \underset{\sim}{w}_i)_\alpha \pm (\underset{\sim}{w}_j')_\alpha (\underset{\sim}{w}_k')_\alpha$$

$$= \frac{dr_j}{dc} \pm \sum_{\bar{\sigma}} \theta_i (\underset{\sim}{w}_j')_\alpha \, (\underset{\sim}{w}_i')_\alpha \pm (\underset{\sim}{w}_j')_\alpha \, (\underset{\sim}{w}_k')_\alpha$$

$$(5.26) \qquad = \frac{dr_j}{dc} \pm (\underset{\sim}{w}_j')_\alpha \sum_{\bar{\sigma}'} \theta_i (\underset{\sim}{w}_i')_\alpha \ ,$$

and as $\sum_{\bar{\sigma}'} \theta_i (\underset{\sim}{w}_i')_\alpha$ is independent of j , the updating of $\frac{dr_j}{dc}$ only requires a single operation.

The operations of equations (5.19) to (5.22), and (5.26) describe the basic updating at change of partition. In practise, the initial step is to find the least squares solution $(\bar{\sigma} = \phi)$. This is done using a Choleski factorization, $LL^T = A^T A$, which also provides the initial P matrix. L is actually lower triangular, but this does not persist beyond the first step.

5.2.3 The Choice of the Orthogonal Transformation

In the updating at a change of partition, an orthogonal transformation, Q , was chosen such that

$$(5.16) \qquad Q^T \underset{\sim}{w}_k = \| \underset{\sim}{w}_k \| \underset{\sim}{e}_\alpha \qquad 1 \le \alpha \le m \ ,$$

without specifying α precisely. We will now show that, as far as $\| \underset{\sim}{w}_j \|$ is concerned, it does not matter which α is chosen.

In (5.19) we had

$$\underset{\sim}{w}_j' = D^{-\frac{1}{2}} Q^T \underset{\sim}{w}_j = D^{-\frac{1}{2}} \underset{\sim}{v}_j \ ,$$

$$(5.27) \qquad \text{where } \underset{\sim}{v}_j = Q^T \underset{\sim}{w}_j \ .$$

Now, in (5.24),

$$D^{-1} = I \mp \frac{\| \underset{\sim}{w}_k \|^2 \, \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T}{1 \pm \| \underset{\sim}{w}_k \|^2}$$

Thus

$$D^{-\frac{1}{2}} = I - \{ 1 \pm (1 \pm \| \underset{\sim}{w}_k \|^2)^{-\frac{1}{2}} \} \underset{\sim}{e}_\alpha \, \underset{\sim}{e}_\alpha^T$$

(5.28) $$= I - (1 \pm \delta) \underset{\sim}{e}_\alpha \underset{\sim}{e}_\alpha^T ,$$

where $\delta = (1 \pm \| \underset{\sim}{w}_k \|^2)^{-\frac{1}{2}}$. Hence

(5.29) $$\underset{\sim}{w}_j' = \underset{\sim}{v}_j - (1 \pm \delta) \underset{\sim}{e}_\alpha (\underset{\sim}{v}_j)_\alpha .$$

So,

$$\| \underset{\sim}{w}_j' \|^2 = \| \underset{\sim}{v}_j \|^2 - 2 (\underset{\sim}{v}_j)_\alpha^2 (1 \pm \delta) + (\underset{\sim}{v}_j)_\alpha^2 (1 \pm \delta)^2$$

$$= \| \underset{\sim}{v}_j \|^2 - (\underset{\sim}{v}_j)_\alpha^2 (1 \pm \delta)(1 \mp \delta)$$

$$= \| \underset{\sim}{v}_j \|^2 - (\underset{\sim}{v}_j)_\alpha^2 (1 - \delta^2)$$

$$= \| \underset{\sim}{v}_j \|^2 - (\underset{\sim}{v}_j)_\alpha^2 \left(1 - \frac{1}{1 \pm \| w_k \|^2} \right)$$

(5.30) $$= \| \underset{\sim}{v}_j \| \mp (\underset{\sim}{v}_j)_\alpha^2 \frac{\| \underset{\sim}{w}_k \|^2}{1 \pm \| w_k \|^2}$$

But, from (5.27), and using (5.20), (5.21), (5.22),

$$(\underset{\sim}{v}_j)_\alpha = (Q \underset{\sim}{w}_j)_\alpha$$

$$= \{ (I - 2 \underset{\sim}{\hat{q}} \, \underset{\sim}{\hat{q}}^T) \underset{\sim}{w}_j \}_\alpha$$

$$= (\underset{\sim}{w}_j)_\alpha - (2 \underset{\sim}{\hat{q}}^T \underset{\sim}{w}_j) \underset{\sim}{\hat{q}}_\alpha$$

$$= (\underset{\sim}{w}_j)_\alpha - \frac{\{(\underset{\sim}{w}_k)_\alpha - \theta_k \| \underset{\sim}{w}_k \| \}\{2\underset{\sim}{w}_k^T \underset{\sim}{w}_k - \theta_k \| \underset{\sim}{w}_k \| (\underset{\sim}{w}_j)_\alpha \}}{\{2(\underset{\sim}{w}_k^T \underset{\sim}{w}_k - \theta_k \| \underset{\sim}{w}_k \| (\underset{\sim}{w}_k)_\alpha \}}$$

$$= (\underset{\sim}{w}_j)_\alpha + \frac{(\underset{\sim}{w}_j)_\alpha \theta_k \| \underset{\sim}{w}_k \| \{(\underset{\sim}{w}_k)_\alpha - \theta_k \| \underset{\sim}{w}_k \| \}}{\| \underset{\sim}{w}_k \| \{ \| \underset{\sim}{w}_k \| - \theta_k (\underset{\sim}{w}_k)_\alpha \}}$$

$$- \frac{\underset{\sim}{w}_j^T \underset{\sim}{w}_k \{(\underset{\sim}{w}_k)_\alpha - \theta_k (\underset{\sim}{w}_k)_\alpha \}}{\| \underset{\sim}{w}_k \| \{ \| \underset{\sim}{w}_k \| - \theta_k (\underset{\sim}{w}_k)_\alpha \}}$$

$$(5.31) \quad = (\underset{\sim}{w}_j)_\alpha - (\underset{\sim}{w}_j)_\alpha + \frac{\underset{\sim}{w}_j^T \underset{\sim}{w}_k \theta_k}{\| \underset{\sim}{w}_k \|} = \frac{\underset{\sim}{w}_j^T \underset{\sim}{w}_k \theta_k}{\| \underset{\sim}{w}_k \|}$$

Substituting (5.31) in (5.30), and using $\| \underset{\sim}{v}_j \| = \| \underset{\sim}{w}_j \|$,

$$(5.32) \quad \| \underset{\sim}{w}'_j \|^2 = \| \underset{\sim}{w}_j \|^2 \mp \frac{(\underset{\sim}{w}_j^T \underset{\sim}{w}_k)^2}{1 \pm \| \underset{\sim}{w}_k \|^2} ,$$

which is independent of α .

The above result shows that $\| \underset{\sim}{w}_i \|$ is independent of which α is chosen. However, from (5.28), (5.29) and (5.31),

$$(5.33) \quad (\underset{\sim}{w}'_j)_\alpha = \frac{\theta_k \underset{\sim}{w}_j^T \underset{\sim}{w}_k}{\| \underset{\sim}{w}_k \| (1 - \| \underset{\sim}{w}_k \|)^{\frac{1}{2}}} ,$$

if, as is normal, $\sigma' = \sigma - \{k\}$.

Now if the same α is chosen at each iteration, and there happened to be a build up of the α^{th} element of some of the $\underset{\sim}{w}_i$, there could be an accelerating process of the $\underset{\sim}{w}_i$ becoming parallel to $\underset{\sim}{e}_\alpha$ and hence to each other. It is therefore suggested that it may be safer to cycle the α . Both methods were tested in the implementation of the algorithm without any apparent difference being found.

5.2.4 Behaviour of Derivatives at Change of Partition

We saw earlier that a change of partition became necessary in order to keep $\underset{\sim}{x}$ feasible, i.e. so that the solution of (5.8) also solved (5.7). We have yet to show that we do end up with a feasible partition for the next range of c, i.e. if $\sigma \to \sigma +' \{k\}$, $\theta_k \frac{dr'_k}{dc} \geq 1$, and if $\sigma \to \sigma - \{k\}$, $\theta_k \frac{dr_k}{dc} < 1$, for if this is not so, then the algorithm could cycle $\sigma \to \sigma \pm' \{k\} \to \sigma \to \ldots$, or end up in an infeasible partition with $\underset{\sim}{x}$ solving (5.8) but not (5.7). This is not simple to show analytically, but comes as a fairly easy consequence of one of the theorems in Chapter 4.

Assume that we have cycling at a change of partition, $\sigma \to \sigma' \to \sigma \to \ldots$, and let $\sigma' = \sigma \cup' \{k\}$. Then we have $\theta_k \frac{dr_k}{dc} \geq 1$, and $\theta_k \frac{dr'_k}{dc} < 1$. If we now reduce c slightly and consider the k^{th} residual, we have $|r_k| \leq c$ and $|r'_k| > c$. This, however, contradicts Theorem 4.4 which states that, at adjacent partitions $|r_k| > c \Leftrightarrow |r'_k| > c$, and $|r_k| \leq c \Leftrightarrow |r'_k| \leq c$.

The above result shows that cycling cannot occur at a simple change of partition, when only one residual changes status. As was shown in example 4.1, however, (discussed in Section 4.2.9), it is possible to have more than one residual changing status at a particular value of c. In this case it is rather more difficult to show that cycling does not occur. This is dealt with in Section 5.3.

5.2.5 The Algorithm Summarised

Algorithm 5.1 To find the M-estimator for any value of c.

Step 1 Perform a Choleski factorisation on the initial matrix and find the least squares estimator.

Step 2 Determine the end of the current range of c and if this includes the final c , calculate the M-estimator and stop.

Step 3 Determine the partition for the next range of c .

Perform the updating required to enable calculation of the M-estimator and residuals within the new partition.

Go to 2 .

5.3 FINITENESS

In order to demonstrate that the algorithm is finite, we have to show that

(i) there are only a finite number of ranges of c ,

(ii) cycling cannot occur at a change of partition.

Theorem 5.2

There are only finitely many ranges of c .

Proof

If we re-write (5.8) as

$$(5.34) \quad \text{minimise } F(c,\underset{\sim}{x}) = \tfrac{1}{2}\sum_{\sigma} r_i^2 + \sum_{\bar{\sigma}_+} (cr_i - \tfrac{1}{2}c^2) - \sum_{\bar{\sigma}_-} (cr_i - \tfrac{1}{2}c^2) \ ,$$

where $\bar{\sigma}_+ = \{i \,|\, r_i(\underset{\sim}{x}^*) > c\}$, $\bar{\sigma}_- = \{i \,|\, r_i(\underset{\sim}{x}^*) < -\, c\}$,

we see that as there are only finitely many partitions σ, $\bar{\sigma}_+$, $\bar{\sigma}_-$, so we need only show that each such partition can only be visited once in order to show (i) above.

Provided B , the submatrix of A defined by σ , is of full rank (and from Theorem 4.6 we know that there is one), the solution of 5.34 is

$$(5.35) \qquad \underset{\sim}{x} = c(B^TB)^{-1}\ B^T(\sum_{\bar{\sigma}_-} \underset{\sim}{a}_i - \sum_{\bar{\sigma}_+} \underset{\sim}{a}_i) \ ,$$

and thus

$$(5.36) \qquad \frac{d\underset{\sim}{x}}{dc} = (B^TB)^{-1}\ B^T(\sum_{\bar{\sigma}_-} \underset{\sim}{a}_i - \sum_{\bar{\sigma}_+} \underset{\sim}{a}_i) \ .$$

Now if the partition $\sigma, \bar{\sigma}_+, \bar{\sigma}_-$ is feasible for c_1 and c_2 , with corresponding solutions $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$, then for any $c_3 = \alpha c_1 + (1-\alpha)c_2$, $0 < \alpha < 1$,

$$\underset{\sim}{x}_3 = \alpha \underset{\sim}{x}_1 + (1-\alpha)\underset{\sim}{x}_2 \text{ is of the form}$$

$$\underset{\sim}{x}_3 = \underset{\sim}{x}_1 - (1-\alpha)\ (c_2-c_1)\ (B^TB)^{-1}B^T(\sum_{\bar{\sigma}_-} \underset{\sim}{a}_i - \sum_{\bar{\sigma}_+} \underset{\sim}{a}_i)$$

$$= \underset{\sim}{x}_1 - \delta c \frac{d\underset{\sim}{x}}{dc} \ .$$

Moreover, for $i \in \sigma, |r_i(\underset{\sim}{x}_1)| \le c$ and $|r_i(\underset{\sim}{x}_2)| \le c \Rightarrow |r_i(\underset{\sim}{x}_3) \le c$

for $i \in \bar{\sigma}_+$, $r_i(\underset{\sim}{x}_1) > c$ and $r_i(\underset{\sim}{x}_2) > c \Rightarrow r_i(\underset{\sim}{x}_3) > c$

for $i \in \bar{\sigma}_-$, $r_i(\underset{\sim}{x}_1) < -c$ and $r_i(\underset{\sim}{x}_2) < -c \Rightarrow r_i(\underset{\sim}{x}_3) < -c$,

so that no residual would want to change status at $\underset{\sim}{x}_3$, i.e. c_1 and c_2 are within the same range of c .

At this point it will be valuable to recall some results from the previous chapter and strengthen a result shown in passing in one of its theorems.

Lemma 4.2 showed that if $\sigma_a = \sigma_b \cup \{k\}$, then for any vector $\underset{\sim}{x}$ satisfying $|r_k(\underset{\sim}{x})| = c$, $F_a(\underset{\sim}{x}) = F_b(\underset{\sim}{x})$.

A partition σ_a was defined as being $\bar{\sigma}$-feasible if $|r_i(\underset{\sim}{x}_a)| > c$, $i \in \bar{\sigma}$.

Theorem 4.2(b) showed that the feasible partition had the smallest function value of all $\bar{\sigma}$-feasible partitions.

Lemma 5.1

If $\sigma_a = \sigma_b \cup \{k\}$, then $F_a(\underset{\sim}{x}_a) \geq F_b(\underset{\sim}{x}_b)$, and if $|r_k(\underset{\sim}{x}_a)| \neq c$, $F_a(\underset{\sim}{x}_a) > F_b(\underset{\sim}{x}_a) \geq F_b(\underset{\sim}{x}_b)$.

Proof

$$\begin{aligned} F_a(\underset{\sim}{x}_a) &= \tfrac{1}{2}\sum_{\sigma_a} r_i(\underset{\sim}{x}_a)^2 + \sum_{\bar{\sigma}_a} (c|r_i(\underset{\sim}{x}_a)| - \tfrac{1}{2}c^2) \\ &= \tfrac{1}{2}\sum_{\sigma_b} r_i(\underset{\sim}{x}_a)^2 + \tfrac{1}{2}r_k(\underset{\sim}{x}_a)^2 + \sum_{\bar{\sigma}_a} (c|r_i(\underset{\sim}{x}_a)| - \tfrac{1}{2}c^2) \\ &\geq \tfrac{1}{2}\sum_{\sigma_b} r_i(\underset{\sim}{x}_a)^2 + c|r_k(\underset{\sim}{x}_a)| - \tfrac{1}{2}c^2 + \sum_{\bar{\sigma}_a} (c|r_i(\underset{\sim}{x}_a)| - \tfrac{1}{2}c^2) \\ &= \tfrac{1}{2}\sum_{\sigma_b} r_i(\underset{\sim}{x}_a)^2 + \sum_{\bar{\sigma}_b} (c|r_i(\underset{\sim}{x}_a)| - \tfrac{1}{2}c^2) \\ &= F_b(\underset{\sim}{x}_a) \end{aligned}$$

with the "$\geq$" being strictly "$>$" unless $|r_k(\underset{\sim}{x}_a)| = c$.

Finally, as $\underset{\sim}{x}_b$ minimises F_b ,

$$F_b(\underset{\sim}{x}_b) \leq F_b(\underset{\sim}{x}_a) \leq F_a(\underset{\sim}{x}_a) .$$

We now show that when there is more than one residual wanting to change status, cycling can be avoided. As in the case of just one residual (Section 5.2.4), we do not consider the partition changes using residual derivatives at c_0 but, equivalently, residuals at a slightly reduced value of c , c_1 , the reduction in c being slight enough so that residuals not equal to c_0 in size when $c = c_0$ are still not equal to c_1 in size when $c = c_1$. We do this basically by giving an algorithm guaranteed to find a feasible partition for a given value of c , starting from an arbitrary partition.

Algorithm 5.2

To find a feasible partition for any value of c , starting from an arbitrary partition and proceeding only by adjacent partition changes, i.e. $\sigma \to \sigma \pm \{k\}$.

Step 1 Starting from any initial partition, change partitions $\sigma \to \sigma \cup \{k\}$, where $|r_k| \leq c$, until a partition σ_1 is reached which is $\bar{\sigma}$-feasible.

Set $i \leftarrow 1$.

Step 2 If σ_i is feasible, stop;
else for any k such that $k \in \sigma_i, |r_k(\underset{\sim}{x}_i)| > c$,
form $\sigma_{i+1} = \sigma_i - \{k\}$.

Set $i \leftarrow i + 1$.

If σ_i is $\bar{\sigma}$-feasible, go to 2 ; else

Step 3 Find $\underset{\sim}{y}_i$ of the form

$$\underset{\sim}{y}_i = \alpha \underset{\sim}{x}_i + (1-\alpha)\underset{\sim}{x}_{i-1} , \quad 0 < \alpha \le 1$$

such that $|r_j(\underset{\sim}{y}_i)| \ge c$, $j \in \bar{\sigma}_i$,

and $|r_k(\underset{\sim}{y}_i)| = c$ for (at least one) $k \in \bar{\sigma}_i$.

Form $\sigma_{i+1} = \sigma_i \cup \{j \,|\, |r_j(\underset{\sim}{y}_i)| = c \}$.

Set $i \leftarrow i + 1$.

If σ_i is $\bar{\sigma}$-feasible, go to 2; else

Step 4 Find $\underset{\sim}{y}_i$ of the form

$$\underset{\sim}{y}_i = \alpha \underset{\sim}{x}_i + (1-\alpha)\underset{\sim}{y}_{i-1} \quad 0 \le \alpha \le 1$$

such that $|r_j(\underset{\sim}{y}_i)| \ge c$ $j \in \bar{\sigma}_i$

and $|r_k(\underset{\sim}{y}_i)| = c$ for (at least one) $k \in \bar{\sigma}_i$.

Form $\sigma_{i+1} = \sigma_i \cup \{j \,|\, |r_j(\underset{\sim}{y}_i)| = c\}$.

If σ_i is $\bar{\sigma}$-feasible, go to 2 ;

Else go to 4 .

Theorem 5.3

Algorithm 5.2 terminates with a feasible partition.

Proof

As, in steps 3 and 4 of the algorithm, the partition change is of the form $\sigma_{i+1} = \sigma_i \cup \{ \quad \}$, it is clear that a sequence of $\bar{\sigma}$-feasible partitions is generated at step 2.

We will prove that this is a finite sequence by showing that the function values of this sequence of $\bar{\sigma}$-feasible partitions decreases monotonically. We will denote by σ_a the last $\bar{\sigma}$-feasible partition found in the sequence.

At step 2, we have $\sigma_{i+1} = \sigma_i - \{k\}$, with $|r_k(\underset{\sim}{x}_i)| \neq c$, so by Lemma 5.1,

$$(5.37) \qquad F_{i+1}(\underset{\sim}{x}_{i+1}) < F_i(\underset{\sim}{x}_i) \equiv F_a(\underset{\sim}{x}_a) ,$$

so that if σ_{i+1} is the next $\bar{\sigma}$-feasible partition in the sequence, the function value has been decreased.

At step 3, by Lemma 4.2 and convexity,

$$(5.38) \qquad F_{i+1}(\underset{\sim}{x}_{i+1}) \leq F_{i+1}(\underset{\sim}{y}_i) = F_i(\underset{\sim}{y}_i) \leq \alpha F_i(\underset{\sim}{x}_i) + (1-\alpha)\, F_i(\underset{\sim}{x}_{i-1}) \leq$$

$$F_i(\underset{\sim}{x}_i-1) .$$

But as, by Lemma 5.1 (remembering that $\sigma_{i-1} \rightarrow \sigma_i$ was the partition change in step 2, so that $\sigma_{i-1} = \sigma_i \cup \{k\}$, and $\sigma_{i-1} \equiv \sigma_a$) ,

$F_{i-1}(\underset{\sim}{x}_{i-1}) > F_i(\underset{\sim}{x}_{i-1}) \geq F_i(\underset{\sim}{x}_i)$,

$$(5.39) \qquad F_{i+1}(\underset{\sim}{x}_{i+1}) \leq F_{i+1}(\underset{\sim}{y}_i) < F_{i-1}(\underset{\sim}{x}_{i-1}) \equiv F_a(\underset{\sim}{x}_a) .$$

Thus, if σ_{i+1} is the next $\bar{\sigma}$-feasible partition in the sequence, the function value has again been decreased.

At step 4, by Lemma 4.2 and convexity,

$$(5.40) \qquad F_{i+1}(\underset{\sim}{x}_{i+1}) \leq F_{i+1}(\underset{\sim}{y}_i) = F_i(\underset{\sim}{y}_i) \leq \alpha F_i(\underset{\sim}{x}_i) + (1-\alpha)\, F_i(\underset{\sim}{y}_{i-1}) \leq$$

$$F_i(\underset{\sim}{y}_{i-1}) .$$

So, by (5.39), (taking into account the incrementing of i that has taken place)

$$(5.41) \qquad F_{i+1}(\underset{\sim}{x}_{i+1}) \leq F_{i+1}(\underset{\sim}{y}_i) \leq F_i(\underset{\sim}{y}_{i-1}) < F_{i-2}(\underset{\sim}{x}_{i-2}) \equiv F_a(\underset{\sim}{x}_a) .$$

At each cycle through step 4, (5.40) and (5.41) still hold (each (5.41) now being proved by (5.40) and the previous (5.41)), so then when finally a $\bar{\sigma}$-feasible partition is reached, its function value is less than that of the previous $\bar{\sigma}$-feasible partition.

Finally we observe that as no $\bar{\sigma}$-feasible partition can be repeated, and there are only finitely many of them, the sequence must terminate, and as the only stop in the algorithm is at a feasible partition, the final partition is feasible.

It is interesting to note here the connection between algorithm 5.2 and Theorem 4.2(b). The theorem stated that of all $\bar{\sigma}$-feasible partitions, the feasible partition has smallest function value. The algorithm generates a sequence of $\bar{\sigma}$-feasible partitions of monotonically decreasing function value, ending with the feasible partition.

Theorem 5.4

It is possible to avoid cycling at a change of partition, even when there is a possibility of more than one residual being involved in changing status.

Proof

Assume we have a feasible partition σ_a at $c = c_a$.

Let $\sigma_= = \{i \,|\, |r_i(\underset{\sim}{x}_a)| = c_a\}$ and let S be the set of all partitions defined by

$$S = \{\sigma | \sigma \subseteq \sigma_a,\ \bar{\sigma} \subseteq \bar{\sigma}_a \cup \sigma_=\}.$$

Let $c_b < c_a$ be such that

(i) $|r_i(\underset{\sim}{x}_j)| > c$, $i \in \bar{\sigma}_a$, $\sigma_j \in S$

(ii) $|r_i(\underset{\sim}{x}_j)| < c$, $i \in \sigma_a - \sigma_=$, $\sigma_j \in S$.

Then, with $c = c_b$, and starting from σ_a , algorithm 5.2 shows how to avoid cycling in reaching a feasible partition. Moreover, from the definition of c_b only partitions belonging to S are involved in the process, i.e. only residuals with indices belonging to $\sigma_=$ are involved in partition changes.

Now because $|r_i| = c_a$ at $c = c_a$, $i \in \sigma_=$, for all partitions belonging to S , and because, from Theorem 5.1, r_i is a linear function of c within any partition, performing partition changes based on residual values at c_b is equivalent to performing partition changes based on residual derivatives at c_a , which completes the proof.

Note that we have not shown above that cycling cannot occur, but rather how it can be avoided. This feature was not built into the implementation of algorithm 5.1, and in the rare cases (all artificially and deliberately constructed, eg. example 4.1, Section 4.3.7) where multiple residual change did occur, a natural treatment coped quite adequately.

Algorithm 5.2 can, of course, be used to calculate the M-estimator if c is known, or, with some modification, if c is to be estimated at the same time. It may be reasonably efficient, as only residual values at "old" partitions were needed to define the next partition, never the residuals at the partition so defined.

This algorithm has not been implemented, being rather similar to Huber's algorithm described earlier. Note, however, that Huber (1973) could not guarantee finiteness for his algorithm.

5.4 NUMERICAL RESULTS

Apart from checking the examples in Chapter 4 which were small enough to calculate by hand and diverse enough to cover most cases, test problems were generated in the following way. An $(n-m-1) \times (m+1)$ matrix was generated using uniformly distributed random numbers. Then an $m \times (m+1)$ matrix was prepended onto it as rows 1 to m, this matrix being chosen so that $\underset{\sim}{x} = \underset{\sim}{1}$ solved the equations $r_i = 0$, $i = 1,\ldots,m$. Finally a row was added so that $\underset{\sim}{\lambda} = \underset{\sim}{\tfrac{1}{2}}$ satisfied the least absolute deviation (LAD) criterion

$$\sum_{i=1,m} \lambda_i \underset{\sim}{a}_i + \sum_{i=m+1,n} \theta_i \underset{\sim}{a}_i = \underset{\sim}{0} ,$$

where the $\underset{\sim}{a}_i$ are m-vectors corresponding to the rows of A, and $\theta_i = \text{sgn}(\sum_{j=1,m} A_{ji} - A_{m+1,i})$. Thus $\sigma_0 = \{1,\ldots,m\}$. Then the program was run on the test data so generated, allowing c to reduce right down to zero, and checking to see that the LAD partition was as obtained as expected.

Results are summarised in Table 5.1. Testing was done on a DEC 10 computer, and times are internal CPU times. In some cases, the final partition had $|\sigma| > m$, although in every such case $\{1,\ldots,m\}$ was a subset of σ_0. In one of the $(m=8, n=100)$ runs, a residual moved from σ to $\bar{\sigma}$, back to σ at a reduced level of c, amd then out to $\bar{\sigma}$ again when c was reduced still further. This was the only occasion in 45 test runs that this happened. The time per iteration was calculated excluding setup time (reading in data and performing the initial Choleski factorisation), which was consistently close to .8n × (iteration time). A facility was built into the program to make $\underset{\sim}{e}_{\alpha}$ of (5.20) either always $\underset{\sim}{e}_1$, or to cycle $\underset{\sim}{e}_1 \to \underset{\sim}{e}_2 \to \dots \to \underset{\sim}{e}_m \to \underset{\sim}{e}_1 \to \dots$ There was no discernable difference in any run using the two approaches, although it is conceivable that with more ill-conditioned data that the cycling approach could avoid problems.

Table 5.1 demonstrates a time/iteration relationship of $0(nm)$ and an $0(n^2m)$ relationship of total time in solving LAD problems. It is not, of course, advocated that this algorithm be used for solving LAD problems, as there clearly must be at least $n-m$ iterations, a complexity greater than that observed elsewhere - 5 runs of 10 × 200 on the LAD algorithm described in Chapter 6, for example, took about 3.5 seconds.

TABLE 5.1

Total time for 5 Test Runs (Seconds)/

Time per iteration (m secs).

n m	50	100	200
6	2.41	9.75	40.32
	10.20	20.15	41.08
8	3.00	12.30	49.95
	12.80	25.27	50.60
10	3.50	14.50	59.21
	15.22	30.25	60.51

5.5 M-ESTIMATOR DUALITY

5.5.1 LP Duality

For an LP problem,

$$\text{minimise } \underset{\sim}{c}^T\underset{\sim}{x} \text{ subject to } A\underset{\sim}{x} \ge \underset{\sim}{b} \text{ ,}$$

the K.T. optimality conditions for $\underset{\sim}{x}^*$ to be optimal are

(5.42) $A\underset{\sim}{x}^* \ge \underset{\sim}{b}$

(5.43) for some $\underset{\sim}{u} \ge \underset{\sim}{0}$, $\underset{\sim}{c} = A^T\underset{\sim}{u}$

(5.44) $\underset{\sim}{u}^T(A\underset{\sim}{x}-\underset{\sim}{b}) = 0$.

These are referred to as primal feasibility, dual feasibility, and complimentary slackness respectively. Now the complimentary slackness condition can be interpreted as an appropriate subproblem being optimised. This is rather trivially true in the case of LP problems, as the appropriate subproblem is constrained to a single point - as indeed it is for every iteration of the simplex algorithm, each point being the vertex of a polyhedron defined by the set of active constraints. To illustrate this point further, consider the RLS algorithm. At each step (including the last) the appropriate subproblem has the form

$$\text{minimise } \tfrac{1}{2}(A\underset{\sim}{x}-\underset{\sim}{b})^2 \text{ subject to } x_i = 0 \text{ , } i \in \sigma \text{ .}$$

The optimality criteria for this subproblem are

(5.45) $x_i = 0$, $i \in \sigma$

(5.46) for some (unsigned) $\underset{\sim}{u}$, $A^TA\underset{\sim}{x} - A^T\underset{\sim}{b} = I\underset{\sim}{u}$

(5.47) $\underset{\sim}{u}^T \underset{\sim}{x} = 0$.

The dual variables $\underset{\sim}{u}$ in (5.46) are also the dual variables for the RLS problem, so (5.47), $\underset{\sim}{u}^T \underset{\sim}{x} = 0$ is also the complimentary slackness condition for the RLS problem.

5.5.2 M-estimator Duality

In this section we wish to point out some parallels between duality as it applies to LP problems and some of the results of the last two chapters.

In LP problems, the optimality criteria are primal feasibility, dual feasibility and complimentary slackness. The optimality criteria for the M-estimator function are σ-feasibility, $\bar{\sigma}$-feasibility and optimisation of the relevant subproblem (i.e. corresponding partition).

It is well known that the optimal function value for LP problems is the smallest primal-feasible value and the largest dual-feasible value. Theorems 4.2(a) and (b) showed that the optimum value for the M-estimator is the smallest of all σ-feasible points and the largest of all $\bar{\sigma}$-feasible points.

The algorithm for the restricted least squares problem in Chapter 3 had an overall pattern of moving towards dual feasibility, keeping complimentary slackness at each iteration. Primal feasibility is ignored until dual feasibility is achieved, when it is used for optimality testing. If the optimality test fails, a procedure is described for finding another dual feasible point with smaller function value, neither primal nor dual feasibility being necessary at intermediate points. In algorithm 5.2, for the M-estimator, the overall pattern is to move towards $\bar{\sigma}$-feasibility, optimising subproblems at each iteration. σ-feasibility is ignored until $\bar{\sigma}$-feasibility is achieved, when it is used for optimality testing. If the optimality test fails, a procedure is described for finding another $\bar{\sigma}$-feasible point with smaller function value, neither σ - nor $\bar{\sigma}$-feasibility being necessary at intermediate points.

The above observations, whilst in no way constituting a duality theory for the M-estimator, at least suggest a close parallel between LP duality and $\sigma/\bar{\sigma}$-feasibility for the M-estimator, and are tentatively advanced as perhaps the first halting steps towards a duality theory for the M-estimator.

CHAPTER 6

AN ALGORITHM FOR THE LEAST ABSOLUTE DEVIATION ESTIMATOR

6.1 INTRODUCTION

The least absolute deviation (LAD) estimator is the solution to the problem

$$(6.1) \qquad \min\Sigma|r_i| \quad , \qquad r_i = \underset{\sim}{a}_i^T \underset{\sim}{x} - b_i \; .$$

The problem predates least squares (LS) estimation, being proposed by Boscovitch in 1757 to fit a line to points on the plane. He also added the constraint that the line should pass through the centroid of the points, and gave a method for finding the line. Edgeworth, in 1887, dropped the constraint and gave a new method for the solution of (6.1). Due, however, to the lack of continuity of the derivative of the modulus function, the problem (6.1) is difficult and despite continuing interest in it, the relative ease of computation of the LS method contributed to that method's greater popularity. An historical review of LAD, and LS, regression can be found in Harter (1974-1976).

The renewed interest in LAD stems largely from the search for robust methods which arose in the early 1970's (eg. Hampel, 1971). Harter (1977) suggested a two-stage method for calculating regressions, using the kurtosis calculated after doing one regression analysis to estimate the kurtosis of the data, after which an L_1, L_2 or L_∞ regression would be done according to whether the data was leptokurtic, mesokurtic or platykurtic. Huber (1974) did point out that among all L_p estimates, only L_1 is technically robust. He also claims (Huber, 1977,b) that it has a rather low asymptotic efficiency in approximately normal situations, and that there are better methods.

Still, the method is extremely robust and over the last decade or so there have been several algorithms developed to solve the problem. Those which have proved efficient fall into two categories; gradient algorithms (Bartels, Conn and Sinclair, 1978, Bloomfield and Steiger, 1979); and algorithms based on linear programming (LP).

Bloomfield and Steiger (1979) attribute to Harris in 1950 the realisation that the LAD problem could be turned into an LP problem. The first practical LP formulation was probably by Barrodale and Young (1966), who expressed each residual as the difference between two non-negative variables. The LP algorithms solve either a primal LP problem using a modified simplex algorithm, (Barrodale and Roberts, 1973, Spyropoulos, Kiountouzis and Young, 1973, Armstrong and Frome, 1976,b), or a dual LP problem by a modified dual simplex algorithm (Robers and Ben-Israel, 1969, Abdelmalek, 1975).

The other approach which has proved successful is based on a theorem of Usow (1967) where he showed that "the set of best L_1 approximations is a closed convex set which is the convex hull of best L_1 approximations to $f(\underset{\sim}{x})$ for which there are at least m zeroes". These methods find a vertex (i.e. m linearly independent zero residuals), decide on one residual to allow to become non-zero, and perform a line-search in the direction so defined.

In a rather startling result, paralleling that by Dixon (1972) for Quasi-Newton algorithms, Osborne (1980) has shown that all of the LP and gradient algorithms mentioned above are identical in that they produce identical sequences of iterations provided they use

(i) equivalent starting procedures

(ii) equivalent procedures for entering and leaving the basis.

The algorithm presented below is essentially a gradient method, but differs from those above in that it does not insist that there must be $m-1$ residuals kept to zero in every descent direction, thereby allowing a greater freedom in the choice of such directions.

6.2 THE ALGORITHM

6.2.1 General Description

The approach described below was suggested by the defining function of the M-estimator. The algorithm described in the previous chapter can, as was observed there, be used to calculate LAD, but suffers from the disadvantages of having at least $n-m$ iterations, and so consequently a lot of downdating, together with starting from a fixed point which may not be a good approximation to the LAD estimator.

Instead of minimising the AD function directly, we consider

$$(6.2) \qquad \min F(\underset{\sim}{x},c) = \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\Sigma|r_i| \ ,$$

where $\underset{\sim}{x}_0$ is an initial estimate, and c is a continuation parameter. We shall show below that for large enough, but finite, c, the minimiser of (6.2) is the LAD estimator.

At the starting point, $c = 0$, and (6.2) is not very difficult to solve. There is still, however, the problem that some residuals may be zero, and given that there will be (at least) m zero residuals at the optimum, we will sometimes want to force some residuals to stay on zero for a range of c . So we re-write (6.2) as

$$(6.3) \qquad \min F(\underset{\sim}{x},c) = \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\sum_{\sigma}\theta_i r_i + c\sum_{\bar{\sigma}}\nu_i r_i \ ,$$

where $\bar{\sigma}$ is the set of indices of residuals which are to stay at zero whilst c is increased. The optimality conditions for $\underset{\sim}{x}$ to solve (6.3) are

$$(6.4) \qquad \underset{\sim}{x} = \underset{\sim}{x}_0 - c\sum_{\sigma}\theta_i \underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\nu_i \underset{\sim}{a}_i \ .$$

Differentiating (6.4) with respect to c ,

$$(6.5) \qquad \frac{d\underset{\sim}{x}}{dc} = -\sum_{\sigma}\theta_i \underset{\sim}{a}_i - \sum_{\bar{\sigma}}\nu_i \underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\underset{\sim}{a}_i \frac{d\nu_i}{dc} \ ,$$

and so

$$(6.6) \qquad \frac{dr_j}{dc} = -\underset{\sim}{a}_j^T\sum_{\sigma}\theta_i \underset{\sim}{a}_i - \underset{\sim}{a}_j^T\sum_{\bar{\sigma}}\nu_i \underset{\sim}{a}_i - c\underset{\sim}{a}_j^T\sum_{\bar{\sigma}}\underset{\sim}{a}_i \frac{d\nu_i}{dc} = 0 \quad \text{for} \quad j \in \bar{\sigma} \ .$$

Letting B correspond to the subset of A defined by $\bar{\sigma}$,

$$B^TB \; \frac{d}{dc} \; (c\underset{\sim}{\nu}) = -B^T\sum_{\sigma}\theta_i\underset{\sim}{a}_i$$

(6.7) $$\frac{d}{dc}\,(c\underset{\sim}{\nu}) = -(B^TB)^{-1} \; B^T\sum_{\sigma}\theta_i\underset{\sim}{a}_i = \underset{\sim}{\lambda} \; .$$

$$\therefore \; c\underset{\sim}{\nu} = c\underset{\sim}{\lambda} \; + \underset{\sim}{\mu}$$

(6.8) $$\underset{\sim}{\nu} = \underset{\sim}{\lambda} + \frac{\underset{\sim}{\mu}}{c} \; ,$$

where $\underset{\sim}{\lambda}$ and $\underset{\sim}{\mu}$ are constant vectors, $\underset{\sim}{\lambda}$ being defined by $\bar{\sigma}$ and $\underset{\sim}{\mu}$ depending on the starting point. Equations (6.3) to (6.6) now become

(6.9) $$\min F(\underset{\sim}{x},c) = \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\sum_{\sigma}\theta_i\underset{\sim}{a}_i + c\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i + \sum_{\bar{\sigma}}\mu_i\underset{\sim}{a}_i$$

(6.10) $$\underset{\sim}{x} = \underset{\sim}{x}_0 - c\sum_{\sigma}\theta_i\underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i - \sum_{\bar{\sigma}}\mu_i\underset{\sim}{a}_i$$

(6.11) $$\frac{d\underset{\sim}{x}}{dc} = -\sum_{\sigma}\theta_i\underset{\sim}{a}_i - \sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i$$

(6.12) $$\frac{dr_j}{dc} = -\underset{\sim}{a}_j^T\sum_{\sigma}\theta_i\underset{\sim}{a}_i - \underset{\sim}{a}_j^T\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i \; .$$

The above equations indicate that $\underset{\sim}{x}$ is a linear function of c . We now wish to determine for which range of c $\underset{\sim}{x}$ as defined by (6.10) solves (6.3).

Theorem 6.1

If $\underset{\sim}{x}$ solves (6.3) so that

(6.4) $$\underset{\sim}{x} = \underset{\sim}{x}_0 - c\sum_{\sigma}\theta_i\underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\nu_i\underset{\sim}{a}_i$$

and $\quad i \in \sigma \Leftrightarrow r_i(\underset{\sim}{x}) \neq 0$,

then $\underset{\sim}{x}$ solves (6.2) iff $|\underset{\sim}{\nu}| \leq \underset{\sim}{1}$, and if it does solve (6.2) it is the unique solution.

Proof

(i) Assume $|\underset{\sim}{\nu}| \leq \underset{\sim}{1}$, but $\underset{\sim}{x}$ does not solve (6.2). Then there exists $\underset{\sim}{\delta}$ such that

$$F(\underset{\sim}{x}+\underset{\sim}{\delta},c) - F(\underset{\sim}{x},c) \leq 0$$

$$\Rightarrow \tfrac{1}{2}(\underset{\sim}{x}+\underset{\sim}{\delta}-\underset{\sim}{x}_0)^2 - \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\sum_{\sigma}\theta_i\underset{\sim}{a}_i^T\underset{\sim}{\delta} + c\sum_{\bar{\sigma}}|\underset{\sim}{a}_i^T\underset{\sim}{\delta}| \leq 0$$

$$\Rightarrow (\underset{\sim}{x}-\underset{\sim}{x}_0)^T\underset{\sim}{\delta} + \underset{\sim}{\delta}^T\underset{\sim}{\delta} + c\sum_{\sigma}\theta_i\underset{\sim}{a}_i^T\underset{\sim}{\delta} + c\sum_{\bar{\sigma}}|\underset{\sim}{a}_i^T\underset{\sim}{\delta}| \leq 0$$

$$\Rightarrow -c\sum_{\sigma}\theta_i\underset{\sim}{a}_i^T\underset{\sim}{\delta} - c\sum_{\bar{\sigma}}\nu_i\underset{\sim}{a}_i^T\underset{\sim}{\delta} + \underset{\sim}{\delta}^T\underset{\sim}{\delta} + c\sum_{\sigma}\theta_i\underset{\sim}{a}_i^T\underset{\sim}{\delta} + c\sum_{\bar{\sigma}}|\underset{\sim}{a}_i^T\underset{\sim}{\delta}| \leq 0$$

$$\Rightarrow c\sum_{\bar{\sigma}}(-\nu_i\underset{\sim}{a}_i^T\underset{\sim}{\delta}+|\underset{\sim}{a}_i^T\underset{\sim}{\delta}|) + \underset{\sim}{\delta}^T\underset{\sim}{\delta} \leq 0 ,$$

a contradiction, as $\nu_i \leq 1$.

(ii) If $\nu_k > 1$ for some k , we merely choose $\underset{\sim}{\delta}$ such that $\underset{\sim}{a}_i^T\underset{\sim}{\delta} = 0$, $\quad i \in \bar{\sigma} - \{k\}$, and $\|\underset{\sim}{\delta}\|$ very small.

We now see that the range of c for which (6.10) solves (6.2) is defined by

(6.13) $\quad$ (i) $\quad \theta_i r_i > 0 \qquad i \in \sigma$

$\qquad\qquad$ (ii) $\quad |c\lambda_i+\mu_i| \leq c \qquad i \in \bar{\sigma}$.

The algorithm thus proceeds from range to range of c, changing the partition $\bar{\sigma} \to \bar{\sigma} \cup \{k\}$ if r_k violates (6.13(i)), or $\sigma \to \sigma \cup \{k\}$ if λ_k and μ_k violate (6.13(ii)). Updating at partition change will be described in Section 6.2.2, but it is worth noting here that if $\bar{\sigma} \to \bar{\sigma} \cup \{k\}$, μ_k is chosen so that $c\lambda_k + \mu_k = c\theta_k$, so that $\underset{\sim}{x}$, and $F(\underset{\sim}{x},c)$ are continuous in c. The algorithm stops when $\bar{\sigma}$ defines a full basis, so that $\frac{d\underset{\sim}{x}}{dc} = 0$ and thus $\frac{dr_j}{dc} = 0$, so that (6.13(i)) can never be violated, and $|\lambda_i| < 1$, $i \in \bar{\sigma}$, so that (6.13(ii)) can never be violated. This corresponds to the optimality criteria for the LAD,

$$\sum_{\sigma} \theta \underset{\sim}{a}_i + \sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i = \underset{\sim}{0}\ , \quad |\lambda_i| \le 1\ , \quad i \in \bar{\sigma}$$

It will be observed that the method is a steepest descent method in that the direction of descent $\frac{d\underset{\sim}{x}}{dc}$ is the projection of the derivative of $\sum_{\sigma} \theta_i r_i$ on the hyperplane defined by $r_i = 0$, $i \in \bar{\sigma}$.

We have yet to show that for large enough, but finite, c, the solution of (6.2) is also a solution of (6.1).

Theorem 6.2(a)

If $\underset{\sim}{x}'$ is the unique solution to the LAD problem (6.1), then for $c \ge c_0$, where c_0 is finite, $\underset{\sim}{x}'$ solves (6.2).

Proof

As $\underset{\sim}{x}'$ is the LAD estimator, we have

$$\sum_{\sigma} \theta_i a_i + \sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i = \underset{\sim}{0}\ ,$$

and, as $\bar{\sigma}$ spans the space, we can find $\underset{\sim}{\mu}$ such that

$$(6.14) \qquad \underset{\sim}{x}' = \underset{\sim}{x}_0 - \sum_{\bar{\sigma}} \mu_i \underset{\sim}{a}_i$$

$$= \underset{\sim}{x}_0 - c\sum_{\sigma} \theta_i \underset{\sim}{a}_i - c\sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i - \sum_{\bar{\sigma}} \mu_i \underset{\sim}{a}_i \, .$$

Now as $\underset{\sim}{x}'$ solves (6.1) uniquely, $|\lambda_i| < 1$, $i \in \bar{\sigma}$, we can choose c large enough so that

$$(6.15) \qquad |c\lambda_i + \mu_i| \le c \, , \quad i \in \bar{\sigma} \, ,$$

so that by Theorem 6.1, $\underset{\sim}{x}'$ (uniquely) solves (6.2).

We now extend the above theorem to the case where the LAD is not unique. As Theorem 6.1 has shown that (6.2) has a unique minimum, not all LAD minima will solve (6.2).

Theorem 6.2(b)

If the LAD is not unique, and S is defined by

$$S = \{ \underset{\sim}{x} | \underset{\sim}{x} \text{ solves (6.1)} \} \, ,$$

and $\underset{\sim}{x}'$ solves

$$\min \tfrac{1}{2} (\underset{\sim}{x} - \underset{\sim}{x}_0)^2 \, , \quad \underset{\sim}{x} \in S \, ,$$

then, for large enough but finite c , $\underset{\sim}{x}'$ solves (6.2).

Proof

In the proof of Theorem 6.2(a), at equation (6.15) we used the fact that $|\lambda_i| < 1$ to state that for c large enough equation (6.15) could be satisfied. We shall show that for $\underset{\sim}{x}'$ if $|\lambda_i| = 1$, then $\mathrm{sgn}\,\mu_i = -\mathrm{sgn}\,\lambda_i$, so that again (6.15) can be satisfied.

With S defined as above, it is well known (Usow, 1967) that S is a polyhedron with vertices having m zero residuals. We shall now parameterise these vertices. Consider any vertex, $\underset{\sim}{y}$. Like every other point within S, it satisfies

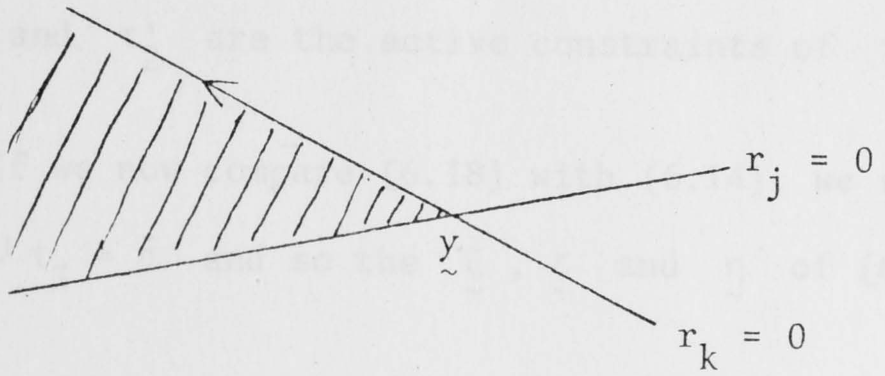
$$(6.16) \qquad \sum_{\sigma} \theta_i \underset{\sim}{a}_i + \sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i = \underset{\sim}{0} \, ,$$

where $\sigma = \{i \,|\, r_i(\underset{\sim}{y}) \neq 0\}$.

Now for any $k \in \bar{\sigma}$ such that $|\lambda_k| = 1$, if we move slightly away from $\underset{\sim}{y}$ to $\underset{\sim}{y}'$ in a direction such that $r_i(\underset{\sim}{y}') = 0$, $i \in \bar{\sigma} - \{k\}$, $\mathrm{sgn}r_k(\underset{\sim}{y}') = \lambda_k$, and close enough to $\underset{\sim}{y}$ so that $\mathrm{sgn}r_i(\underset{\sim}{y}') = \mathrm{sgn}r_i(\underset{\sim}{y})$, $i \in \sigma$, then we still have, at $\underset{\sim}{y}'$,

$$\sum_{\sigma \cup \{k\}} \theta_i \underset{\sim}{a}_i + \sum_{\bar{\sigma} - \{k\}} \lambda_i \underset{\sim}{a}_i = \underset{\sim}{0} \, ,$$

and so $\underset{\sim}{y}'$ is within S



For all other edges away from $\underset{\sim}{y}$, $r_k = 0$, so that the hyperplane $r_k = 0$ supports S with $\mathrm{sgn}r_k(\underset{\sim}{y}') = \mathrm{sgn}\lambda_k$ for all $\underset{\sim}{y}' \in S$. We now define

$$\tau_1 = \{i \mid \lambda_i = 1 \text{ at some vertex of } S\}$$

$$\tau_2 = \{i \mid \lambda_i = -1 \text{ at some vertex of } S\}$$

$$\tau_3 = \{i \mid |\lambda_i| < 1 \text{ at all vertices of } S\} .$$

It is clear that τ_1, τ_2 and τ_3 are discrete sets, their union summing to $\bar{\sigma}$, and that for all points within S , where (6.16) holds, if $\lambda_i = 1$, $i \in \tau_1$, and if $\lambda_i = -1$, $i \in \tau_2$.

With S thus re-defined, we see that $\underset{\sim}{x}'$ solves

$$(6.17) \qquad \min \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2$$

$$\text{subject to} \quad \underset{\sim}{a}_i^T\underset{\sim}{x} - b_i \geq 0 \quad i \in \tau_1$$
$$-\underset{\sim}{a}_i^T\underset{\sim}{x} + b_i \geq 0 \quad i \in \tau_2$$
$$\underset{\sim}{a}_i^T\underset{\sim}{x} - b_i = 0 \quad i \in \tau_3 .$$

From the Kuhn-Tucker conditions we have $\exists\ \underset{\sim}{\xi} \geq \underset{\sim}{0}, \underset{\sim}{\zeta} \geq \underset{\sim}{0}, \underset{\sim}{\eta}$ unsigned such that

$$(6.18) \qquad \underset{\sim}{x} - \underset{\sim}{x}_0 = \sum_{\tau_1'} \xi_i\underset{\sim}{a}_i - \sum_{\tau_2'} \zeta_i\underset{\sim}{a}_i + \sum_{\tau_3} \eta_i\underset{\sim}{a}_i$$

where τ_1' and τ_2' are the active constraints of τ_1 and τ_2.

If we now compare (6.18) with (6.14), we see that, at $\underset{\sim}{x}'$ $\tau_1' \cup \tau_2' \cup \tau_3 = \bar{\sigma}$ and so the $\underset{\sim}{\xi}$, $\underset{\sim}{\zeta}$ and $\underset{\sim}{\eta}$ of (6.18) are the $\underset{\sim}{\mu}$ of (6.14)

$$\text{i.e.} \quad \mu_i = -\xi_i \leq 0 \qquad i \in \tau_1'$$
$$\mu_i = \zeta_i \geq 0 \qquad i \in \tau_2'$$
$$\mu_i = \eta_i \qquad i \in \tau_3 .$$

But we had, for $i \in \tau_1$, $\lambda_i = +1$, and for $i \in \tau_2$, $\lambda_i = -1$. So we see that in either case $|c\lambda_i + \mu_i| \leq c$, and so (6.15) can be satisfied even if $|\lambda_i| = 1$. This completes the proof.

6.2.2 Updating at Change of Partition

Once, for a particular range of c, the correct partitioning σ, $\bar{\sigma}$ has been determined and the signs of r_i, $i \in \sigma$ known, then $\underset{\sim}{\lambda}$ can be calculated from (6.7)

$$\underset{\sim}{\lambda} = -(B^T B)^{-1} B^T \sum_{\sigma} \theta_i \underset{\sim}{a}_i .$$

This then determines $\underset{\sim}{\mu}$, for in ensuring that $\underset{\sim}{x}$ and therefore r_i are continuous, we need, for $\sigma \to \sigma - \{k\}$, that $c\lambda'_k + \mu'_k = \text{sgn } r_k$, and for $\sigma \to \sigma \pm \{k\}$, that $c\underset{\sim}{\lambda}' + \underset{\sim}{\mu}' = c\underset{\sim}{\lambda} + \underset{\sim}{\mu}$.

Once $\underset{\sim}{\lambda}$ and $\underset{\sim}{\mu}$ are known, we can determine $\underset{\sim}{x}$ for any value of c from (6.10), $\frac{d\underset{\sim}{x}}{dc}$ from (6.11), $\frac{dr_j}{dc}$ from (6.12) and the limits of this range from (6.13). The updating of $\underset{\sim}{\lambda}$ is therefore basic and we now describe an efficient and stable method of doing this.

Basically we shall use a QU factorisation of B, where Q is orthogonal, $Q^T Q = I$, and U is upper triangular. (6.7) then becomes

$$(6.19) \qquad \underset{\sim}{\lambda} = -U^{-1} Q^T \sum_{\sigma} \theta_i \underset{\sim}{a}_i .$$

If $\bar{\sigma}' = \bar{\sigma} \cup \{k\}$, $B' = (B, \underset{\sim}{a}_k)$ and in this case the orthogonalisation is performed using the Gram-Schmidt process. If we let

$$Q' = (Q, \underset{\sim}{q}) , \quad U' = \begin{pmatrix} U, & \underset{\sim}{r} \\ \underset{\sim}{0}^T, & \rho \end{pmatrix} ,$$

we require $\underset{\sim}{q}$, $\underset{\sim}{r}$ and ρ to satisfy

(6.20) $\quad Q'U' = B'$,

(6.21) $\quad Q^T\underset{\sim}{q} = \underset{\sim}{0}$, $\underset{\sim}{q}^T\underset{\sim}{q} = 1$.

Expanding (6.20), we have

$$(B,\underset{\sim}{a}_k) = (Q,\underset{\sim}{q}) \begin{pmatrix} U & \underset{\sim}{r} \\ \underset{\sim}{0}^T & \rho \end{pmatrix}$$

$$= QU, Q\underset{\sim}{r} + \rho\underset{\sim}{q} \; .$$

So

(6.22) $\quad Q\underset{\sim}{r} + \rho\underset{\sim}{q} = \underset{\sim}{a}_k$.

Multiplying (6.22) by Q^T ,

(6.23) $\quad \underset{\sim}{r} = Q^T\underset{\sim}{a}_k$.

Then from (6.22) and (6.23),

(6.24) $\quad \rho\underset{\sim}{q} = (I-QQ^T)\underset{\sim}{a}_k$,

and as $\|\underset{\sim}{q}\| = 1$, we also have ρ and $\underset{\sim}{q}$.

If $\bar{\sigma}' = \bar{\sigma} - \{k\}$, and k is the last column of B , Q' and U' are merely the first $k - 1$ columns of Q and U . If k is not the last column of B , we re-order B so that it is, using Givens matrices. A Givens matrix is a symmetrical orthogonal matrix of the form $G = \begin{pmatrix} \gamma & \delta \\ \delta & -\gamma \end{pmatrix}$.

If we had

$$(B_1, \underset{\sim}{a}_k, B_2) = Q \begin{pmatrix} U_1 & \underset{\sim}{r} & U_{12} \\ \underset{\sim}{0}^T & \rho & \underset{\sim}{u}^T \\ 0 & \underset{\sim}{0} & U_2 \end{pmatrix} ,$$

then

$$(B_1, B_2) = Q \begin{pmatrix} U_1 & U_{12} \\ \underset{\sim}{0}^T & \underset{\sim}{u}^T \\ 0 & U_2 \end{pmatrix} = Q(R_1, R_2) = Q\hat{R} .$$

The matrix R is upper Hessenberg, eg if $p = |\bar{\sigma}| = 5$ and $k = 3$,

$$\hat{R} = \begin{pmatrix} X & X & X & X \\ 0 & X & X & X \\ 0 & 0 & X & X \\ 0 & 0 & X & X \\ 0 & 0 & 0 & X \end{pmatrix} .$$

If we now choose Givens matrices $G_{k,k+1}, \ldots, G_{p-1,p}$ so that

$$G\hat{R} = G_{p-1,p} \cdots G_{k,k+1} \hat{R} = \begin{pmatrix} U' \\ \underset{\sim}{0}^T \end{pmatrix} , \text{ then}$$

$$QG^T = QG_{k,k+1} \cdots G_{p-1,p} = (Q', \underset{\sim}{q})$$

has orthonormal columns and

$$(B_1, B_2) = B' = Q'U' \text{ as required.}$$

The above treatments are given in some detail by Daniel, Gragg, Kaufman and Stewart (1976). In that same paper, they consider the possibility that in the Gram-Schmidt process $Q^T\underset{\sim}{q} \neq \underset{\sim}{0}$. They suggest a test

$$(6.25) \qquad \|\underset{\sim}{a}_k\| + \omega\|Q^T\underset{\sim}{q}\| < \theta\,\|\underset{\sim}{q}\|\ ,$$

where ω and θ are termination parameters. If the above test fails, they re-orthogonalise, using the most recently computed $\underset{\sim}{q}$ instead of $\underset{\sim}{a}_k$, both in the Gram-Schmidt process and in the test (6.25). After some experimentation, they have suggested $\omega = 0$, $\theta = 1.4$ for the termination parameters. If the test fails more than four times consecutively, they initialise re-start procedures. In the implementation of the algorithm, the re-orthogonalisation procedure was used, but four failures of the test (6.25) was treated as degeneracy in the model and the observed value, b_k , of that observation perturbed. This only ever happened with specifically constructed examples, never in the main testing using randomly generated data.

6.2.3 Progress of the Algorithm

In this section we attempt to give an over-view of what happens within the algorithm as c is increased. Specifically, we establish three results about the progress of the algorithm: that $F(c)$ is concave in c ; that $\|\underset{\sim}{x}-\underset{\sim}{x}_0\|$ does not decrease as c increases; that $\Sigma|r_i|$ does not increase as c increases.

Theorem 6.3

$$F(c) = \min \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\Sigma|r_i| \quad \text{is concave in } c\ .$$

Proof

$$F(c) = \tfrac{1}{2}(\underset{\sim}{x}_c-\underset{\sim}{x}_0)^2 + c\sum_{\sigma}\theta_i r_i + c\sum_{\bar{\sigma}}\lambda_i r_i + \sum_{\bar{\sigma}}\mu_i r_i$$

with $$\frac{d\underset{\sim}{x}_c}{dc} = -c\sum_{\sigma}\theta_i\underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i = -(\underset{\sim}{\alpha}+\underset{\sim}{\beta})$$

$$\frac{dF(c)}{dc} = (\underset{\sim}{x}_c - \underset{\sim}{x}_0)^T \frac{d\underset{\sim}{x}_c}{dc} + \sum_{\sigma} \theta_i r_i + c\sum_{\sigma} \theta_i \underset{\sim}{a}_i^T \frac{d\underset{\sim}{x}_c}{dc} + c\sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i \frac{d\underset{\sim}{x}_c}{dc}$$

$$+ \sum_{\bar{\sigma}} \lambda_i r_i + \sum_{\bar{\sigma}} \mu_i \underset{\sim}{a}_i^T \frac{d\underset{\sim}{x}_c}{dc}$$

$$= \sum_{\sigma} \theta_i r_i + \sum_{\bar{\sigma}} \lambda_i r_i$$

(6.26) $$= \sum_{\sigma} \theta_i r_i = \sum |r_i|$$

Differentiating (6.26) with respect to c ,

(6.27) $$\frac{d^2F(c)}{dc^2} = \sum_{\sigma} \theta_i \underset{\sim}{a}_i^T \frac{d\underset{\sim}{x}}{dc} = - \underset{\sim}{\alpha}^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) .$$

Now $\frac{dr_j}{dc} = 0 , j \in \bar{\sigma}$

i.e. $\underset{\sim}{a}_j^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) = 0 , j \in \bar{\sigma}$

$\therefore \sum_{\bar{\sigma}} \lambda_i \underset{\sim}{a}_i^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) = 0$

(6.28) i.e. $\underset{\sim}{\beta}^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) = 0$.

So, from (6.27) and (6.28)

$$\frac{d^2F(c)}{dc^2} = -\underset{\sim}{\alpha}^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) - \underset{\sim}{\beta}^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta})$$

$$= -(\underset{\sim}{\alpha}+\underset{\sim}{\beta})^T (\underset{\sim}{\alpha}+\underset{\sim}{\beta}) \le 0 .$$

So $\frac{d^2F(c)}{dc^2}$ is piecewise constant, but always ≤ 0 . Thus $F(c)$ is concave in c .

Corollary

$$\Sigma |r_i| = \frac{dF}{dc} \text{ is non-increasing.}$$

Theorem 6.4

$\|\underset{\sim}{x}_c - \underset{\sim}{x}_0\|$ does not decrease as c increases.

Proof

Within a range of c, where $\underset{\sim}{x}_c$ and r_j are linear functions of c,

$$\frac{dF}{dc} = (\underset{\sim}{x}_c - \underset{\sim}{x}_0)\frac{d\underset{\sim}{x}_c}{dc} + c\frac{d}{dc}\Sigma|r_i| + \Sigma|r_i|$$

$$= \Sigma|r_i| \text{ from } (6.26) .$$

$$\therefore \quad (\underset{\sim}{x} - \underset{\sim}{x}_0)\frac{d\underset{\sim}{x}}{dc} = -c\frac{d}{dc}\Sigma|r_i|$$

$$= -c\frac{d^2F}{dc^2}, \text{ from } (6.26).$$

Hence,

$$\frac{d}{dc}\tfrac{1}{2}(\underset{\sim}{x}_c - \underset{\sim}{x}_0)^2 = (\underset{\sim}{x} - \underset{\sim}{x}_0)\frac{d\underset{\sim}{x}}{dc}$$

$$= -c\frac{d^2F}{dc^2}$$

$$\geq 0 \text{ from Theorem 6.3 .}$$

So that within a range of c, $\|\underset{\sim}{x}_c - \underset{\sim}{x}_0\|$ does not decrease as c increases, and as $\underset{\sim}{x}_c$ is continuous, this applies to all c.

Note that in the above results we had $\Sigma|r_i|$ non-increasing, and $\|\underset{\sim}{x}_c - \underset{\sim}{x}_0\|$ non-decreasing, rather than $\Sigma|r_i|$ decreasing and $\|\underset{\sim}{x}_c - \underset{\sim}{x}_0\|$ increasing. Normally the $\underset{\sim}{a}_i$, $i \in \bar{\sigma}$, will not span the space and then $\underset{\sim}{\alpha} + \underset{\sim}{\beta} \neq \underset{\sim}{0}$, so that $\|\underset{\sim}{\alpha} + \underset{\sim}{\beta}\| > 0$. However, it can occur that the $\underset{\sim}{a}_i$, $i \in \bar{\sigma}$, do span the space at a sub-optimal partition, when $\frac{d\underset{\sim}{x}}{dc} = \underset{\sim}{0}$ so that neither $\Sigma|r_i|$ nor $||\underset{\sim}{x}_c - \underset{\sim}{x}_0||$ will change during that range of c .

Example 6.1

4	2	1	1	0	3	2
1	3	1	0	4	7	5
5	5	4	.999	2	18	6

For $.063 \le c \le .064$, $\bar{\sigma} = \{1,4\}$, $\underset{\sim}{x}_c^T = (.999, 1.004)$,

but the LAD has $\bar{\sigma} = \{1,2\}$, $\underset{\sim}{x}_c^T = (1,1)$.

6.2.4 The Algorithm Summarised

Algorithm 6.1 To find the LAD estimator.

Step 1 Find an initial estimate, and the initial derivatives $\frac{d\underset{\sim}{x}}{dc}$, $\frac{dr_i}{dc}$.

Step 2 Determine, c_m , the end of the current range of c . If

```
        c_m = ∞ ,  stop;
else
        determine the new partitioning  σ' ,  σ̄';
        amend  $\underset{\sim}{\lambda}$ , $\underset{\sim}{\mu}$ , $\frac{d\underset{\sim}{x}}{dc}$ , $\frac{dr_i}{dc}$ ;
        go to 2 .
```

6.3 REFINEMENTS TO THE ALGORITHM

The limits to a range of c were given in (6.13) as

(i) $\theta_i r_i \ge 0$ $i \in \sigma$

(ii) $|c\lambda_i + \mu_i| \le c$ $i \in \bar{\sigma}$.

Consider now the situation where $\bar{\sigma}' = \bar{\sigma} \cup \{k\}$, and $|\lambda_k| > 1$. Provided that intervening adjustments to $\bar{\sigma}$ do not cause $|\lambda_k|$ to become ≤ 1 , eventually condition (ii) will be violated and k will move out of $\bar{\sigma}$ again.

Approaching the same question from another viewpoint, the aim of the algorithm is to find a full basis for which $|\lambda_i| \leq 1$, $i \in \bar{\sigma}$. If we consider this final basis and isolate the last two elements, j and k , say,

$$(B,\underset{\sim}{a}_j,\underset{\sim}{a}_k) = (Q,\underset{\sim}{q}_j,\underset{\sim}{q}_k) \begin{pmatrix} U & \underset{\sim}{r}_j & \underset{\sim}{r}_k \\ \underset{\sim}{0}^T & \rho_j & \rho_{jk} \\ \underset{\sim}{0}^T & 0 & \rho_k \end{pmatrix} ,$$

then

$$\lambda_k = \frac{\underset{\sim}{q}_k^T\underset{\sim}{\alpha}}{\rho_k}$$

$$\lambda_j = \frac{\underset{\sim}{q}_j^T\underset{\sim}{\alpha} - \lambda_k\rho_{jk}}{\rho_j} = \frac{\underset{\sim}{q}_j^T\underset{\sim}{\alpha} - \lambda_k\underset{\sim}{q}_j^T\underset{\sim}{a}_k}{\rho_j} .$$

If now k is dropped from $\bar{\sigma}$,

$$\lambda_j' = \frac{\underset{\sim}{q}_j^T(\underset{\sim}{\alpha} - \operatorname{sgn}\lambda_k\underset{\sim}{a}_k)}{\rho_j} ,$$

and comparing the two λ_j 's , we see that, given $|\lambda_k| \leq 1$, they are not too different, so knowing that $|\lambda_j| \leq 1$ it is likely that $|\lambda_j'| \leq 1$. And this applies to all subsets of $\bar{\sigma}$.

There are exceptions to both arguments, of course, but together they do provide an indication that if $|\lambda_k| > 1$ that k is probably not a good candidate for inclusion in the final $\bar{\sigma}$. It therefore seems a good heuristic not to introduce k into $\bar{\sigma}$, merely leaving it in σ but changing the sign of r_k. The factorisation QU will then not have to be performed, and $\underset{\sim}{\lambda}$, $\underset{\sim}{\mu}$, $\frac{d\underset{\sim}{x}}{dc}$ and $\frac{dr_j}{dc}$ will only have to be updated once, rather than twice.

The necessary condition for the resulting partitioning σ_+, σ_-, $\bar{\sigma}$ to be feasible is that if k is left in σ and $\theta'_k = -\theta_k$ (remember $r_k = 0$ at the point in question), then $\mathrm{sgn}\frac{dr'_k}{dc} = \mathrm{sgn}\frac{dr_k}{dc}$. Now,

$$\frac{dr'_k}{dc} = -\underset{\sim}{a}_k^T(\underset{\sim}{\alpha}'+\underset{\sim}{\beta}')$$

$$= -\underset{\sim}{a}_k^T(\underset{\sim}{\alpha}' - QQ^T\underset{\sim}{\alpha}')$$

$$= -\underset{\sim}{a}_k^T(\underset{\sim}{\alpha}-2\theta_k\underset{\sim}{a}_k-QQ^T\underset{\sim}{\alpha} + 2\theta_kQQ^T\underset{\sim}{a}_k)$$

$$(6.29) \qquad = \frac{dr_k}{dc} + 2\theta_k(\underset{\sim}{a}_k^T\underset{\sim}{a}_k - \underset{\sim}{a}_k^TQQ^T\underset{\sim}{a}_k) ,$$

and as $\mathrm{sgn}\frac{dr_k}{dc} = -\theta_k$, for $\mathrm{sgn}\frac{dr'_k}{dc}$ to equal $\mathrm{sgn}\frac{dr_k}{dc}$, we need

$$(6.30) \qquad \left|\frac{dr_k}{dc}\right| > 2(\underset{\sim}{a}_k^T\underset{\sim}{a}_k - \underset{\sim}{a}_k^TQQ^T\underset{\sim}{a}_k) .$$

This condition, (6.30), is equivalent to the condition $|\lambda_k| > 1$ if k is introduced into $\bar{\sigma}$, for

$$|\lambda_k| > 1$$

$$\Rightarrow \left|\frac{(\underset{\sim}{\alpha}-\theta_k\underset{\sim}{a}_k)^T\underset{\sim}{q}}{\rho_k}\right| > 1$$

$$\Rightarrow \left|\frac{(\underset{\sim}{\alpha}-\theta_k\underset{\sim}{a}_k)^T\ (I-QQ^T)\underset{\sim}{a}_k}{\underset{\sim}{a}_k^T\underset{\sim}{a}_k - \underset{\sim}{a}_k^TQQ^T\underset{\sim}{a}_k}\right| > 1$$

$$\Rightarrow \left|\frac{-dr_k}{dc} - \theta_k\underset{\sim}{a}_k^T(I-QQ^T)\underset{\sim}{a}_k\right| > \underset{\sim}{a}_k^T\underset{\sim}{a}_k - \underset{\sim}{a}_k^T\,QQ^T\underset{\sim}{a}_k ,$$

and as $\operatorname{sgn} \frac{dr_k}{dc} = -\theta_k$, this condition becomes (6.30).

In terms of what is happening in the algorithm, the above modification is equivalent to (hypothetically) adjusting the starting point, for we had

$$\underset{\sim}{x} = \underset{\sim}{x}_0 - c\sum_{\sigma}\theta_i\underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i - \sum_{\bar{\sigma}}\mu_i\underset{\sim}{a}_i$$

$$= (\underset{\sim}{x}_0 - 2c\theta_k\underset{\sim}{a}_k) - c\sum_{\sigma}\theta_i'\underset{\sim}{a}_i - c\sum_{\bar{\sigma}}\lambda_i\underset{\sim}{a}_i - \sum_{\bar{\sigma}}\mu_i\underset{\sim}{a}_i$$

where $\theta_i' = \theta_i$, $i = k$, $\theta_k' = -\theta_k$.

The above modification gave mark II of the algorithm. Mark IIIcame from not adjusting $\frac{d\underset{\sim}{x}}{dc}$, the descent direction, when a residual changed sign, but minimising the function along the line $\frac{d\underset{\sim}{x}}{dc}$, in other words continuing along $\frac{d\underset{\sim}{x}}{dc}$ so long as it is a descent direction, although only until the first residual changes sign will it be the steepest descent direction. As each residual changes sign, the only updating necessary is to $\underset{\sim}{\alpha}$, all other vectors can be updated at the minimum along the line. The minimum along the line is recognised by condition (6.23), but $\frac{dr_k}{dc}$ will not have been updated. Rather than amend all $\frac{dr_i}{dc}$ at each change of sign of residual, $\frac{dr_k}{dc}$ can be calculated directly for the test from equation (6.12).

6.4 FINITENESS

As in the case of the M-estimator, in order to establish the finiteness of the algorithm we need to show

(i) there is only a finite number of ranges of c

(ii) cycling cannot occur at partition changes at the end of a range.

Theorem 6.5

If $\underset{\sim}{x}_1$ and $\underset{\sim}{x}_2$ solve (6.3)

$$\min \quad F(\underset{\sim}{x}) = \tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{x}_0)^2 + c\sum_{\sigma}\theta_i r_i + c\sum_{\bar{\sigma}}\nu_i r_i$$

for values of c of c_1 and c_2 respectively, and have the same sign pattern, i.e.

$$r_i(\underset{\sim}{x}_1) = 0 \Leftrightarrow r_i(\underset{\sim}{x}_2) = 0$$

$$\operatorname{sgn} r_i(\underset{\sim}{x}_1) = \operatorname{sgn} r_i(\underset{\sim}{x}_2) ,$$

and if they also solve (6.2),

i.e. $-\underset{\sim}{1} \le \underset{\sim}{\nu}' \le \underset{\sim}{1}$, $-\underset{\sim}{1} \le \underset{\sim}{\nu}^2 \le \underset{\sim}{1}$,

then c_1 and c_2 are in the same range of c .

Proof

$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 - c_1\sum_{\sigma}\theta_i \underset{\sim}{a}_i - c_1\sum_{\bar{\sigma}}\nu_i' \underset{\sim}{a}_i$$

$$\underset{\sim}{x}_2 = \underset{\sim}{x}_0 - c_2\sum_{\sigma}\theta_i \underset{\sim}{a}_i - c_2\sum_{\bar{\sigma}}\nu_i^2 \underset{\sim}{a}_i$$

Let $\quad \underset{\sim}{x}_3 = \alpha \underset{\sim}{x}_1 + (1-\alpha)\underset{\sim}{x}_2$ for any $0 \le \alpha \le 1$

$$c_3 = \alpha c_1 + (1-\alpha)c_2$$

$$\underset{\sim}{\nu}^3 = \frac{1}{c_3}\{\alpha c_1 \underset{\sim}{\nu}' + (1-\alpha)c_2\underset{\sim}{\nu}^2\} .$$

Then $\quad c_3\nu_i^3 = \alpha c_1 \nu_j' + (1-\alpha)c_2\nu_i^2$

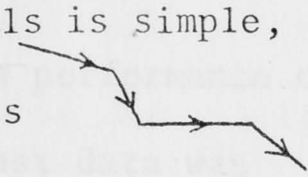
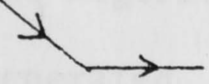
But $\quad -1 \le \nu_i' , \nu_i^2 \le 1$, so

$$-(\alpha c_1 + (1-\alpha)c_2) \le c_3\nu_i^3 \le \alpha c_1 + (1-\alpha)c_2$$

$$-1 \le \nu_i^3 \le 1 ,$$

and so by Theorem 6.1, $\underset{\sim}{x}_3$ uniquely solves (6.2) for $c = c_3$.

The above theorem is sufficient to establish the first condition for finiteness. The second, that cycling cannot occur (or can be avoided) is more difficult. In the M-estimator case only two things could happen, a residual could become larger in size than c without changing sign, or it could become smaller than c in size. Here, three things can happen. A residual can stay zero, change sign or remain the same sign, and the problem is correspondingly more complex. In practise, given the rarity of such an occurrence, no special provision was made for such a contingency, but largely for housekeeping reasons whenever a residual was about to become non-zero it was given a very small value of the appropriate sign, and this is enough to prevent cycling.

On the question of the complexity of the algorithm it is difficult to say much at all. Intuitively, one expects that the number of iterations remaining would be tied to the sign pattern of the residuals. Certainly, the behaviour of most residuals is simple, $\frac{dr_i}{dc}$ rarely changing sign, so that their behaviour is  or  Example 6.1, however, illustrates what can happen. The LAD estimator is $\underset{\sim}{x}^T = (1,1)$, with residuals $(0,0,-2,.001,2,-8,1)$. If we use the LS estimate as our starting point, $\underset{\sim}{x}_0^T = (1.30, 1.32)$, the initial residuals are $(1.53, 1.56, -1.38, .30, 3.28, -4.85, 3.20)$ and, apart from $\bar{\sigma}$, the sign patterns are identical. However, in the algorithm r_4 becomes zero first, stays zero, and then becomes positive again.

6.5 NUMERICAL RESULTS AND DISCUSSION

Initial testing was done on test data generated as described in Section 5.4, and initial results were rather encouraging. Mark II performed better than mark I, and mark III better still, so that a set of experiments on 10 variable, 900 observation data averaged only 13 iterations and about 3 sec execution time on the DEC 10 computer - and this without an efficient sorting algorithm being used in the line - search.

Further testing, however, did not present as rosy a picture. The algorithm was tested on the same type of data used by Bloomfield and Steiger (1979) in comparing their algorithm against that of Barrodale and Roberts. This used Pareto distributions with sometimes infinite variance, sometimes finite but certainly long-tailed. On this data, the algorithm fared badly taking over 200 iterations to do a single 10 variable, 900 observation example. One pleasing feature, though, was that there was no appearance of numerical instability even in these examples.

In attempting to explain the discrepancy in performance of the algorithm, we look more closely at the way the test data was generated. The first set was generated by filling the matrix A with independent uniformly distributed random numbers in the range $-\frac{1}{2}$ to $\frac{1}{2}$, and the vector $\underset{\sim}{b}$ with independent uniformly distributed random numbers in the range $-{}^{m}/2$ to ${}^{m}/2$, followed by minor adjustments to make the LAD easily identifiable. In fitting a model

$$\underset{\sim}{b} = A\underset{\sim}{x} + \underset{\sim}{\varepsilon}$$

to $\underset{\sim}{b}$ and A, it is not clear what the distribution of $\underset{\sim}{\varepsilon}$ would be, but it would certainly be long-tailed, making it a suitable test for an L_1 model. It is suggested, however, that the contours of $\Sigma|r_i|$ with such data would be roughly circular, whereas the contours of the Pareto distribution with its very much greater spread of numbers would be very elliptical. Now is is a well-known property of the steepest descent method in the continuous case that it performs well if the contours are circular, but given long narrow sloping valleys it tends to zig-zag across them.

It would appear that allowing more freedom in the choice of descent direction merely allows zig-zagging to occur, whereas in algorithms such as Bloomfield-Steiger, only one residual is freed from zero to define the search direction, so that it is constrained to move down the valleys. There is confirmation of this in an unpublished result of M.J.D. Powell where he showed that for deterministic problems the strategy of moving off just one residual at a time can be shown to be near-optimal in a certain sense.

Before, however, discarding the freer approach entirely, a couple of variations should be tried. The first is to keep track of which residuals have changed sign during a line-search, and if any try to change sign again in the next line-search to stop at that point and make the next search direction keep that residual at zero, even where $|\lambda_k| > 1$. The second is to try to do the equivalent of a conjugate gradient method, working out a direction in the manner of the Fletcher-Reeves method, and then using a projection of this, rather than the gradient, on the hyperplane of $r_i = 0$, $i \in \bar{\sigma}$. Both of these variations are left for further investigation.

BIBLIOGRAPHY

ABDELMALEK, N.N. (1975) "An efficient method for the discrete linear L_1 approximation problem". Math. Comp. 29, 844-850.

ANDREWS, D.F. (1971) "Significance tests based on residuals". Biometrika 58, 139-148.

ANSCOMBE, F.J. (1960) "Rejection of Outliers". Technometrics 2, 123-147.

ARMSTRONG, R.D. and FROME, E.L. (1976,a) "A branch-and-bound solution of a restricted least squares problem". Technometrics 18, 447-450.

——— (1976,b) "A comparison of two algorithms for absolute deviation curve fitting". J. Amer. Stat. Assoc. 71, 326-330.

BARRODALE, I. and ROBERTS, F.D.K. (1973) "An improved algorithm for discrete L_1 linear approximation". SIAM J. Numer. Anal. 10, 839-848.

BARRODALE, I. and YOUNG, A. (1966) "Algorithms for best L_1 and L_∞ linear approximations on a discrete set". Numerische Mathematik 8, 295-306.

BARTELS, R.H. (1975) "Constrained least squares, quadratic programming, complimentary pivot programming and duality". Technical report No. 218, Dept. Math. Sc., John Hopkins Univ., Baltimore, Ma.

BARTELS, R.H., CONN, A.R. and SINCLAIR, J.W. (1978) "Minimisation techniques for piecewise differentiable functions: The L_1 solution to an overdetermined linear system". SIAM J. Numer. Anal. 15, 224-241.

BARTELS, R.H., GOLUB, G.H. and SAUNDERS, M.A. (1970) "Nonlinear Programming". (Academic Press) 123-176.

BEATON, A.E. and TUKEY, J.W. (1974) "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data". Technometrics 16, 147-185.

BESSEL, F.W. and BAYER, J.J. (1838) "Gradmessungin in Ostpreussen", Berlin. Quoted in Anscombe, 1960.

BEST, M.J. (1975) "A feasible conjugate direction method to solve linearly constrained optimisation problems". J. Opt. Th. Applic. 16, 25-38.

BLOOMFIELD, P. and STEIGER, W. (1979) "Least absolute deviations curve-fitting". Technical report No. 137, Series 2 Dept. of Stats., Princeton University. Sept. 1977, revised Sept. 1979.

BOSCOVITCH, R.J. (1757) Quoted in Harter, 1977.

BYRD, R.H. and PYNE, D.A. (1979) "Some results on the convergence of the iteratively reweighted least squares algorithm for robust regression". Proceedings of the statistical computing section, Amer. Statist. Assoc. 87-90.

CLARK, D.I. (1980) "An algorithm for solving the restricted least squares problem". J. Austral. Math. Soc. B 21, 345-356.

CLARK, D.I. and OSBORNE, M.R. (1980) "On the implementation of a subset selection algorithm for the restricted least squares problem". J. Austral. Math. Soc. B 22, 2-11.

COTTLE, R.W. (1968) "The principal pivoting method of quadratic programming". Mathematics of the Decision Sciences, Part 1. (Amer. Math. Soc., Providence R.I.) 144-162.

COTTLE, R.W. and DANTZIG, G.B. (1968) "Complimentary pivot theory of mathematical programming". Ibid. 115-136.

DANIEL, J.W., GRAGG, W.B., KAUFMAN, L. and STEWART, G.W. (1976) "Reorthogonalisation and stable algorithms for updating the Gram-Schmidt QR factorisation". Math. Comput. 30, 772-795.

DIXON, L.C.W. (1972) "Quasi-Newton algorithms generate identical points". Math. Programming 2, 383-387.

DUFFIN, R.J., PETERSON, E.L. and ZENER, C.M. (1967) "Geometric Programming" (John Wiley, New York).

DUTTER, R. (1977) "Algorithms for the Huber estimator in multiple regression". Computing 18, 167-176.

EDGEWORTH, F.Y. (1887) "A new method of reducing observations relating to several quantities". Phil. Mag. (Fifth Series), 24, 222-223. Quoted by Bloomfield and Steiger, 1979.

EFFROYMSON, M.A. (1960) "Numerical Methods for Digital Computers". (John Wiley) 191-203.

FISHER, R. (1920) Quoted by Huber, 1977a.

GENTLEMAN, J.F. and WILK, M.B. (1975) "Detecting Outliers II". Biometrics 31, 387-410.

GILL, P.E., GOLUB, G.H., MURRAY, W. and SAUNDERS, M.A. (1974) "Methods for modifying matrix factorisations". Math. Comput. 28, 505-535.

GOLUB, G.H. (1965) "Numerical methods for solving linear least squares problems". Numer. Math. 7, 206-216.

GOLUB, G.H. and SAUNDERS, M.A. (1969) "Linear least squares and quadratic programming". Tech. Rep. CS 134, Computer Science Dept., Stanford University, Calif.

GOLUB, G.H. and WILKINSON, J.H. (1966) "Note on the iterative refinement of least squares solutions". Numer. Math. 9, 139-148.

HADLEY, G. (1962) "Linear Programming". (Addison-Wesley, Reading, MA).

HAMPEL, F.R. (1971) "A general qualitative definition of robustness". Ann. Math. Statist. 42, 1887-1896.

———— (1973) "Robust estimation: A condensed partial survey".

Z. Wahrscheinlichkeitstheorie und Verw. Gebiete 27, 87-104.

——— (1974,a) "Rejection rules and robust estimates of location: An analysis of some Monte Carlo results". Proc. European Meeting of Statisticians and 7th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes, Prague 1974.

——— (1974,b) "The influence curve and its role in robust estimation". J. Amer. Statist. Assoc. 69, 383-393.

HARTER, H.L. (1974-76) "The method of least squares and some alternatives ... Parts I-VI". Int. Statist. Rev. 42, 147-74, 259-64, 282; 43, 1-44, 125-90, 269-78; 44, 113-59.

——— (1977) "Nonuniqueness of Least Absolute Values Regression". Comm. Statist. - Theor. Math. A6(9), 829-838.

HASKEL, K.H. and HANSON, R.J. (1978) "An algorithm for linear least squares problems with equality and non-negativity constraints". Sandia Laboratories, SAND 77-0552; June, 1978.

HOFFMAN, K. (1977) "Robust alternatives to the least squares estimator". Math. Operationsforsch. Statist. 8(3), 305-311.

HOLLAND, P.W. and WELSCH, R.E. (1977) "Robust regression using iteratively reweighted least squares". Comm. Statist.-Theor. Math. A6(9), 813-827.

HUBER, P.J. (1972) "Robust statistics: a review". Ann. Math. Statist. 43, 1041-1067.

——— (1973) "Robust regression: asymptotics, conjectures and Monte Carlo". Ann. Statist. 1, 799-821.

——— (1974) "Comment on adaptive robust procedures". J. Amer. Statist. Assoc. 69, 926-927.

——— (1977,a) "Robust Statistical Procedures". SIAM Regional

Conference Series in Applied Mathematics, 27.

———— (1977,b) "Robust methods of estimation of regression coefficients". Math. Operationsforsch. Statist. 8(1), 41-53.

HUBER, P.J. and DUTTER, R. (1974) "Numerical solution of robust regression problems". COMPSTAT 1974 Proceedings of the Symposium on Computational Statistics, 165-172.

JENNINGS, L.S. and OSBORNE, M.R. (1974) "A direct error analysis for least squares". Numer. Math. 22, 325-332.

LAWSON, C.L. and HANSON, R.J. (1974) "Solving Least Squares Problems". (Prentice Hall, N.J.).

LEMKE, C.E. (1968) "On complimentary pivot theory". Mathematics of the Decision Sciences, Part 1. (Amer. Math. Soc., Providence, R.I.), 95-114.

LUENBERGER, D.G. (1973) "Introduction to Linear and Non-linear Programming". (Addison-Wesley, Reading, MA).

McKEAN, J.W. and HETTMANSPERGER, T.P. (1977) "A robust alternative based on ranks to least squares in analysing linear models". Technometrics 19, 275-284.

MICKEY, M.R., DUNN, O.J. and CLARK, V. (1967) "Note on the use of stepwise regression in detecting outliers". Computers and Biomedical Research, 1, 105-111.

ORTEGA, J. and RHEINBOLDT, W. (1970) "Iterative Solution of Nonlinear Equations in Several Variables". (Academic Press, N.Y.).

OSBORNE, M.R. (1976) "On the computation of stepwise regressions". Austral. Computer J. 8, 61-68.

———— (1980) "Descent methods for discrete L_1 problems". Unpublished result.

PEIRCE, B. (1852) "Criterion for the rejection of doubtful observations". Astronomical J. 2, 161-163. Quoted by Anscombe, 1960.

ROBERS, P.D. and BEN-ISRAEL, A. (1969) "An interval programming algorithm for discrete L_1 approximation problems". J. Approx. Theory 2, 323-336.

SCHLOSSMACHER, E.J. (1973) "An iterative technique for absolute deviations curve fitting". J. Amer. Statist. Assoc. 68, 857-865.

STOER, J. (1971) "On the numerical solution of constrained least squares problems". SIAM J. Numer. Anal. 8, 382-411.

SPYROPOULOS, K., KIOUNTOUZIS, E. and YOUNG, A. (1973) "Discrete approximation in the L_1 norm". Computer J. 16, 180-186.

TUKEY, J.W. (1960) "A survey of sampling from contaminated distributions". Contributions to Probability and Statistics, I. Olkin, ed., Stanford University Press, Stanford, CA.

USOW, K.H. (1967) "On L_1 approximation II: Computation for discrete functions and discretization effects". SIAM J. Numer. Anal. 4, 233-244.

WATERMAN, M.S. (1974) "A restricted least squares problem". Technometrics 16, 135-136.

WATSON, G. (1980) "Approximation Theory and Numerical Methods". (John Wiley, N.Y.).

WILKINSON, J.H. (1961) "Error analysis of direct methods of matrix inversion". J. Assoc. Comp. Mach. 8, 281-330.

———— (1965) "Error analysis of transformations based on the use of matrices of the form $I - 2\underset{\sim}{w}\underset{\sim}{w}^H$". Error in Digital

Computation, Vol II, L.B. Rall, ed. (John Wiley, N.Y.), 77-101.

WILKINSON, J.H. and REINSCH, C. (1971) "Linear Algebra". (Springer-Verlag, Berlin).

WOLFE, P. (1959) "The simplex method of linear programming". Econometrica 27, 382-398.

page 58, replaces lines -6 to page 59, line 7 with: "Now in the proof of Theorem 4.1(a), the only feasibility assumptions made on $\underset{\sim}{x}_a$ and $\underset{\sim}{x}_b$ were that for $i \varepsilon S$, $|r_i(\underset{\sim}{x}_a)| > c$ and $|r_i(\underset{\sim}{x}_b)| \leq c$. Thus as $\sigma_b = \sigma_c \cup S_b$ and, for $i \varepsilon S_b$, $|r_i(\underset{\sim}{x}_a)| > c$ and $|r_i(\underset{\sim}{x}_b)| \leq c$, we have $F_c(\underset{\sim}{x}_b) \leq F_c(\underset{\sim}{x}_a)$. Again, as $\sigma_a = \sigma_c \cup S_a$ and, for $i \varepsilon S_a$, $|r_i(\underset{\sim}{x}_b)| > c$ and $|r_i(\underset{\sim}{x}_a)| \leq c$, we have $F_c(\underset{\sim}{x}_a) \leq F_c(\underset{\sim}{x}_b)$. Thus $F_c(\underset{\sim}{x}_a) = F_c(\underset{\sim}{x}_b)$. However, if some $i \varepsilon S_a$,

$$|r_i(\underset{\sim}{x}_a)| < c, \quad F_c(\underset{\sim}{x}_a) < F_c(\underset{\sim}{x}_b),$$ a contradiction."

page 59, line 11 : replace "ϕ" with "φ"

page 60, line 8 : replace "4.10" with "4.10, p52"

page 61, line -1 : replace "$F(\underset{\sim}{x})$" with "$F(\underset{\sim}{x}_1)$

page 63, line 2 : replace "4.4" with "4.3 and Theorem 4.1(b)"

page 63, line 3 : replace "follows." with "follows from Lemma 4.4"

page 67, line 6 : replace "Watson, 1980" with "Watson, 1980, p118"

page 67, lines 9 and 10 : replace "dx" with "dc"

page 98, line -6: replace "small" with "small, and $\underset{\sim}{a}_k^T \underset{\sim}{\delta} > 0$. Then the positive $\underset{\sim}{\delta}^T \underset{\sim}{\delta}$ will be swamped by the negative $(-\nu_k + 1)\ \underset{\sim}{a}_k^T \underset{\sim}{\delta}$"

page 98, line -2 : replace "o" with "c"

page 112, line -9: insert "When working with a full basis" before "The"

page 112, line -8 : replace "6.23" with "6.30"

CORRIGENDA and ADDENDA

Finite Algorithms for Linear Optimisation Problems by David I. Clark

page 3, line -4 : replace "$\Sigma\rho(A\underset{\sim}{x}_i - b_i)$" with "$\Sigma\rho(A\underset{\sim}{x} - \underset{\sim}{b})_i$"

page 7, lines -6 and -3 : replace "complimentary" with "complementary"

page 8, lines 2 and -3 : replace "complimentary" with "complementary"

page 32, line -1 : replace "$U_{11}\ U_{12} \dots U_{1n}$" with "$u_{11}\ u_{12} \dots u_{1n}$"

page 33, line 2 : replace "$\begin{vmatrix} U_{11} & \\ U_{12} & \\ . & U_2^{(2)T} \\ . & \\ U_{1n} & \end{vmatrix}$" with "$\begin{vmatrix} u_{11} & 0 & \dots & 0 \\ u_{12} & x & \dots & x \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ u_{1n} & x & \dots & x \end{vmatrix}$"

page 42, lines 14 and -6 : replace "y_i" with "b_i"

page 54, line 4 : replace "$i \in \sigma$, F_a is strictly convex" with "$i \in \bar{\sigma}$, F_a has a unique minimum"

page 54, line 5 : replace "$\bar{\sigma}$" with "σ"

page 55, line -4 : replace "$r_i(\underset{\sim}{x}^*)^2$" with "$r_i(\underset{\sim}{x}^*)\ \underset{\sim}{a}_i$"

page 56, line -4 : replace "ϕ" with "$\emptyset$"

page 56, line -9 : after "can" insert ", after re-ordering S,"

page 56, after line 13 insert "and $\alpha_i = 1 - \theta_i/\theta_{i+1}$"

page 57, after line 9 insert "Note that if $\alpha_i=1$, then $\theta_i=0$ and $\underset{\sim}{y}_i = \underset{\sim}{x}_b$
and from lemma 4.3 $\underset{\sim}{y}_i$ minimises F_i , so that
$F_i(\underset{\sim}{x}_b) = F_i(\underset{\sim}{y}_i) \leq \underset{\sim}{f}_i(\underset{\sim}{y}_{i+1})$"

page 58, line 7 : replace "σ" with "σ_c"

page 58, line 8 : replace "$\bar{\sigma} = \bar{\sigma}_a \cap \sigma_b$"
with "$\sigma_c = \sigma_a \cap \bar{\sigma}_b$"

page 58, line 9 : replace "$\bar{\sigma} = \bar{\sigma}_b \cap \sigma_a$"
with "$\bar{\sigma}_c = \sigma_b \cap \bar{\sigma}_a$"

.../2