A LINEAR PROGRAMMING APPROACH TO A SIMPLE LINEAR REGRESSION PROBLEM WITH LEAST ABSOLUTE VALUE CRITERION

by

R. D. Armstrong
M. T. Kung

September 1979

This research was partly supported by Project NRO47-021, ONR Contracts N00014-75-C-0616 and N00014-75-C-0569 with the Center for Cybernetic Studies, The University of Texas at Austin. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, TX 78712
(512) 471-1821
A LINEAR PROGRAMMING APPROACH TO A SIMPLE LINEAR REGRESSION PROBLEM WITH LEAST ABSOLUTE VALUE CRITERION

R. D. ARMSTRONG
Associate Professor
Operations Research and Statistics
Graduate School of Business
University of Texas
Austin, Texas 78712

and

M. T. KUNG
Research Associate
Center for Cybernetic Studies
University of Texas
Austin, Texas 78712
ABSTRACT

This paper presents a linear programming approach to solve simple linear regression problems with the least absolute value criterion. The solution technique uses linear programming with an extended minimum ratio rule. A computational study indicates the efficiency of the algorithm.

KEY WORDS

Simple Linear Regression Problem
Least Absolute Value Regression Problem
Goal Programming
Linear Programming
1. Introduction

The simple linear regression problem arises from a fundamental model of statistical analysis. The model consists of an independent (also known as predictor) random variable which is used to determine the value of the dependent (or response) random variable. The simple linear regression fit has been widely used in statistical and economic forecastings. The simple linear regression problem is to find the linear equation which will fit the data comprising of these two variables.

The simple regression problem with a least absolute value criterion has the following form.

\[ \text{Minimize} \sum_{i=1}^{n} |y_i - \alpha - x_i \beta| \]

where \((x_i, y_i), i=1, 2, \ldots, n\) are the observed values.

2. Algorithm

Problem (1) is equivalent to the following linear programming problem [see 4]:

\[ \text{Minimize} \sum_{i=1}^{n} (P_i + N_i) \]

subject to \(\alpha + x_i \beta + P_i - N_i = y_i, \quad i=1, 2, \ldots, n\)
\(P_i \geq 0\) and \(N_i \geq 0, \quad i=1, 2, \ldots, n\)

where \(P_i\) and \(N_i\) are, respectively, the positive and negative deviation associated with the \(i\)-th observation.

The dual problem of (2) is:

\[ \text{Maximize} \sum_{i=1}^{n} \pi_i y_i \]
We shall exploit the structure of (3) to solve this problem with a dual simplex algorithm. This process is identical to solving the problem (2) with the primal simplex algorithm. The following presentation develops a special purpose algorithm using the revised simplex method on the primal problem (2) with a multiple pivot strategy. This strategy enables the method to perform a pivot through several bases in one iteration.

Initially we choose two observations \((x_c, y_c)\) and \((x_d, y_d)\) such that \(x_c \neq x_d\). Hence, the current basis for the LP problem is:

\[
X_B = \begin{pmatrix} 1 & x_c \\ 1 & x_d \end{pmatrix}
\]

The current right hand side is:

\[
Y_B = \begin{pmatrix} y_c \\ y_d \end{pmatrix}
\]

By the adjoint form of the inverse, the initial basis inverse is given as follows:

\[
X_B^{-1} = \frac{1}{x_c - x_d} \begin{pmatrix} -x_d & x_c \\ 1 & -1 \end{pmatrix}
\]

The solution of (2) can be calculated:

\[
\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = X_B^{-1} Y_B
\]

\[
\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{x_c - x_d} \begin{pmatrix} -x_d & x_c \\ 1 & -1 \end{pmatrix} \begin{pmatrix} y_c \\ y_d \end{pmatrix}
\]
Let NB represent the index set of the nonbasic rows. The deviations for (2), or the reduced costs for (3), are given by:

\[ d_i = y_i - \alpha - \beta x_i \]

Define \( \pi_i = \text{sign}(d_i), i \in \text{NB} \). In the computer code, the value assigned to \( \pi_i \) when \( i \in \text{NB} \) and \( d_i = 0 \) is arbitrarily defined to be +1 and, thereafter, the value is determined by the steps of the algorithm. The situation where \( d_i = 0 \) and \( i \in \text{NB} \) corresponds to the case of degeneracy in linear programming and can be resolved as described by Charnes [3]. The details of this procedure will not be discussed here.

The nonbasic dual variables are either +1 or -1 depending on the sign of \( d_i, i \in \text{NB} \). The values of the basic dual variables \( \pi_c \) and \( \pi_d \) are:

\[
\begin{pmatrix}
\pi_c \\
\pi_d
\end{pmatrix} =
\begin{pmatrix}
-\sum_{i \in \text{NB}} \pi_i x_i \\
-\sum_{i \in \text{NB}} \pi_i x_i
\end{pmatrix}
\begin{pmatrix}
\alpha \\
\beta
\end{pmatrix}
\begin{pmatrix}
\sum_{i \in \text{NB}} \pi_i x_i
\end{pmatrix}
\begin{pmatrix}
-1
\end{pmatrix}
\]

Since this is a primal algorithm, the optimality condition for (2) is dual feasibility, namely, \(-1 \leq \pi_i \leq 1, i = c, d\).

If \(|\pi_c| > 1\), the basic dual variable, \( \pi_c \), will leave the basis. If \(|\pi_c| \leq 1\), and \(|\pi_d| > 1\), the algorithm interchanges the indices \( c \) and \( d \).
Thus the variable to leave the basis will always be $\pi_c$. Define $\rho = \text{sign} (\pi_c)$. The value of $\rho$ indicates if $\pi_c$ is to be increased or decreased. A value, $\rho = +1$ if $\pi_c$ is to be decreased and $\rho = -1$ if $\pi_c$ is to be increased.

The algorithm then determines a nonbasic dual variable to enter the basis. The procedure is to take the minimum value from a list of ratios:

$$
\theta_i = \frac{y_i - \alpha - Bx_i}{\rho \xi_i} \quad \text{for } \pi_i \rho \xi_i > 0, \quad i \in \text{NB},
$$

where $\xi_i = \frac{x_i - x_d}{x_c - x_d}$, $i \in \text{NB}$.

Suppose $\theta_s$ is the smallest ratio value, then $\pi_s$, $s \in \text{NB}$, is the nonbasic dual variable to be examined. Firstly, the algorithm checks if $\pi_s$ will enter the basis at a dual feasible level by the following criterion:

$$
|\pi_c| - 2|\xi_s| \leq 1.
$$

If condition (5) is satisfied, $\pi_s$ will enter the basis and $\pi_s$ will be dual feasible. The algorithm assigns $c$ to be the current value of $d$. Then, $d$ is given the value of $s$. For example, if $\pi_3$ and $\pi_5$ are currently basic, and $\pi_3$ is leaving the basis, and if the incoming nonbasic variable is $\pi_{11}$, the new basic variables will be $\pi_5$ and $\pi_{11}$, respectively.

On the other hand, if condition (5) is not satisfied, $\pi_s$ will remain as a nonbasic dual variable because by bringing $\pi_s$ into the basis, $\pi_s$ will still be dual infeasible for (2). Rather, $\pi_s$ will switch from its current bound value to its opposite bound value. Moreover the value of the basic dual variable $\pi_c$ will be increased (or decreased) by

$$
2\rho|\xi_s|.
$$

The algorithm then eliminates $\theta_s$ from the list of ratios.
and examines the next possible candidate to enter the basis from (4).

The algorithm repeats the above procedure until \(-1 \leq \pi_i \leq +1\), \(i=c,d\) is satisfied. It may be noted that to check this condition after the first iteration only \(\pi_c\) need be examined.

3. Steps of the algorithm

In this section, we summarize the algorithm by giving a step-by-step description. New notation, such as \(D, T_1, T_2,\) and \(T\), are introduced to make the algorithm easier to follow.

1. Initialization:

Choose two observations \((x_c, y_c)\) and \((x_d, y_d)\) such that \(x_c \neq x_d\).

Set \(D = \frac{1}{x_c - x_d}\)

\(\alpha = (-x_dy_c + x_cy_d)D\)

\(\beta = (y_c - y_d)D\)

\(\pi_i = \text{sign}(y_i - \alpha - \beta)\) \(i \neq c,d\)

\(\pi_c = \pi_d = 0\)

\(T_1 = \sum_{i=1}^{n} \pi_i\)

\(T_2 = \sum_{i=1}^{n} \pi_i x_i\)

\(T = (x_d T_1 - T_2)D\)

2. If \(|T| > 1\), go to step 4. Otherwise, set \(D = -D\), interchange \(\pi_c\) and \(\pi_d\) by setting \(u+c, c+d, d+u\), continue.

3. \(T = (T_2 - x_d T_1)D\)

If \(|T| > 1\), go to step 4. Otherwise, stop. The current solution is optimal.
4. Set \( \rho = \text{sign} (T) \)

Determine the minimum from the following list of ratios:

\[
\theta_s = \frac{y_i - \alpha - \beta x_i}{\rho (x_i - x_d) D}
\]

for \( \rho (x_i - x_d) D \pi_i > 0 \)

Let \( \theta_s \) be the smallest ratio determined by the minimum ratio test.

5. If \( |T| - 2 |(x_s - x_d)D| \leq 1 \) go to step 7. Otherwise, proceed to step 6.

6. \( T = T - 2 \rho |(x_s - x_d)D| \)

\[
\begin{align*}
\pi_s &= \pm \pi_s \\
T_1 &= T_1 + 2 \pi_s \\
T_2 &= T_2 + 2 \pi_s x_s
\end{align*}
\]

Eliminate \( \theta_s \) from the list of ratios and go to step 4.

7. \( \pi_c = -\rho \)

\[
\begin{align*}
T_1 &= T_1 - \rho - \pi_s \\
T_2 &= T_2 - \rho x_c - \pi_s x_s \\
\pi_s &= 0
\end{align*}
\]

\( \pi_d \) will replace \( \pi_c \) and \( \pi_s \) will replace \( \pi_d \) by setting \( c + d, d + s \).

Set \( D = \frac{1}{x_c - x_d} \)

\[
\begin{align*}
\alpha &= (-x_d y_c + x_c y_d) D \\
\beta &= (y_c - y_d) D
\end{align*}
\]

Go to step 3.

4. Computation Experience

A computational study was carried out to compare the FORTRAN code, LONESL, developed by Sadovski [5] and the FORTRAN code, SIMLP, [1] utilizing the algorithm presented there. Both are special purpose codes
designed expressly to provide least absolute value estimates for a simple linear regression model. The University of Texas CDC 6600 was used in this study. The observations have been drawn from various uniform and normal distributions using a random number generator. The results of the study are summarized in Table 1.

Our study has indicated that this specialization of the linear programming approach first developed by Barrodale and Roberts [2] is uniformly faster than the Sadovski's approach on all problem sizes. In problems with more than 300 observations, the SIMLP is approximately 50 times faster than LONESL. Considerably less storage is required in SIMLP when compared to the Barrodale and Roberts' code and approximately the same amount when compared to the Sadovski's code. Also, Sposito [6] has shown that LONESL may not always converge, while SIMLP utilizes the convergent properties of linear programming theory. Another feature of SIMLP is that there is no accumulative roundoff error present since all necessary values are recalculated from the original data at each iteration.

5. Conclusion

This paper presents a special purpose algorithm to solve simple least absolute value regression problems. The approach utilizes the characteristics and convergent properties of linear programming. With the addition of the multiple pivot strategy in linear programming, the simple least absolute value regression is solved efficiently. From the computational results, it is shown that the code presented here is superior to the published code of Sadovski, in terms of solution time. A listing of the computer code will be found in [1].
Table 1 (A Comparison between SIMLP and LONESL)

<table>
<thead>
<tr>
<th>Number of observations</th>
<th>SIMLP</th>
<th>LONESL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (CPU milliseconds)</td>
<td>Number of iterations</td>
</tr>
<tr>
<td>50</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td>150</td>
<td>46</td>
<td>5</td>
</tr>
<tr>
<td>200</td>
<td>102</td>
<td>5</td>
</tr>
<tr>
<td>250</td>
<td>84</td>
<td>2</td>
</tr>
<tr>
<td>300</td>
<td>70</td>
<td>3</td>
</tr>
<tr>
<td>350</td>
<td>262</td>
<td>3</td>
</tr>
<tr>
<td>400</td>
<td>265</td>
<td>3</td>
</tr>
<tr>
<td>450</td>
<td>212</td>
<td>5</td>
</tr>
<tr>
<td>500</td>
<td>184</td>
<td>3</td>
</tr>
</tbody>
</table>
References


This paper presents a linear programming approach to solve simple linear regression problems with the least absolute value criterion. The solution technique uses linear programming with an extended minimum ratio rule. A computational study indicates the efficiency of the algorithm.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Linear Regression Problem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Least Absolute Value Regression Problem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goal Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Programming</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>