⟨Research Paper⟩

Journal of the Korean
Statistical Society
Vol. 16, No. 2, 1987

A Robust Estimation Procedure for the Linear Regression Model

Bu Yong Kim*

ABSTRACT

Minimum L_1 norm estimation is a robust procedure in the sense that it leads to an estimator which has greater statistical efficiency than the least squares estimator in the presence of outliers. And the L_1 norm estimator has some desirable statistical properties. In this paper a new computational procedure for L_1 norm estimation is proposed which combines the idea of reweighted least squares method and the linear programming approach. A modification of the projective transformation method is employed to solve the linear programming problem instead of the simplex method. It is proved that the proposed algorithm terminates in a finite number of iterations.

1. Introduction

We consider the problem of estimating the parameters of a linear regression model,

$$y = X\beta + \varepsilon \tag{1.1}$$

where y is an n-dimensional vector of observations, X is an $n \times m$ matrix of known quantities on m independent variables, β is a vector containing m unknown parameters, and ε is a vector of n random errors which are independently and identically distributed according to some distribution.

The method of least squares is usually regarded as the most suitable technique for estimating the parameters of a linear regression model when the errors follow a normal distribution and the regressors are orthogonal. In many practical situations, however,

* ADD, P.O. Box 35, Taejon, Korea

it may happen that some model assumptions fail to hold. Furthermore, the assumption of normality of errors in a regression model can not always be taken for granted. When the distributions of errors are non-normal, the least squares method may yield relatively poor estimates of the regression coefficients. A particular problem is estimation in the presence of outliers. If one has a problem with any significant number of outliers, then the underlying error distribution may not be normal, and it has been shown that the least squares method is rather inadequate. This fact has led to the development of an alternative approach which is called minimum L_1 norm estimation.

Statistical properties of the L_1 norm estimators in small samples are not known exactly. However, a number of simulation studies have shown that the L_1 norm estimators are more efficient than least squares estimators in small samples when the errors follow certain heavy-tailed distributions. These studies on the statistical efficiency have been done by Rice and White (1964), Blattberg and Sargent (1971), Kiountouzis (1973), Rosenberg and Carlson (1977), and Pfaffenberger and Dinkel (1978). Bassett and Koenker (1978) provide the asymptotic results on the L_1 norm estimator, and statistical inferences based on their results have been proposed by Dielman and Pfaffenberger (1982).

There are several types of procedures for computing L_1 norm estimates. One type of procedure makes use of linear programming; some examples are Wagner (1959), Fisher (1961), Barrodale and Roberts(1973), Armstrong *et al.* (1979), and Abdelmalek (1980). The others use the reweighted least squares method or optimization techniques; some examples are Schlossmacher(1973), Bartels *et al.* (1978), and Wesolowsky (1981).

In this paper, attention is focused on the development of a new computational procedure for L_1 norm estimation.

2. Proposed Computational Procedure

We consider the following linear programming problem to compute L_1 norm estimates of the parameters in the linear regression model (1.1):

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{l}'\boldsymbol{e}^{+}+\boldsymbol{l}'\boldsymbol{e}^{-} \\ &\text{subject to} \quad \boldsymbol{X\beta}+\boldsymbol{Ie}^{+}-\boldsymbol{Ie}^{-}=\boldsymbol{y} \\ &\qquad \boldsymbol{e}^{+},\boldsymbol{e}^{-}\geq \boldsymbol{0} \end{aligned} \tag{2.1}$$

β unrestricted

where $l'=(1,\cdots,1)$, and I is the n-th order identity matrix. The components of vector e^+ are vertical deviations above the fitted hyperplane and the components of vector e^- are vertical deviations below.

Karmarkar (1984) proposes a new polynomial-time algorithm for linear programming problem of standard form. It is assumed that the minimum of the objective function is zero and there exists an initial feasible solution to the problem with all components positive. And then a projective transformation and its inverse are employed to get an optimal solution. Karmarkar states that any type of linear programming problem can be converted to the standard form by a preprocess, and his approach can be applied to the problem with unknown optimal objective values. One variant of Karmarkar's original algorithm has been proposed by Vanderbei *et al.* (1985). It considers the general linear programming problem which has full rank and an initial interior feasible point. It uses a linear scaling transformation rather than a projective transformation at each iteration, and repeats until some suitable criterion is satisfied. And there is no requirement that the optimal value of the objective function be zero.

The L_1 norm estimation problem can be solved by applying Karmarkar's algorithm or a varient of that to the primal formulation(2. 1). However, it has been shown that the direct application of these algorithms to the primal linear programming problem is not very efficient because the dimension of the constraint matrix of problem(2. 1) becomes large when a large number of observations are involved. So we start with the dual formulation. From the primal formulation, the dual problem can be written as

$$\begin{aligned} &\text{maximize} \quad \mathbf{y}'\mathbf{w} \\ &\text{subject to} \quad X'\mathbf{w}=\mathbf{0} \qquad (2.2) \\ &\qquad\qquad -\mathbf{l}\leq\mathbf{w}\leq\mathbf{l} \end{aligned}$$

where X has full rank, and $\mathbf{w}=(w_1, w_2, \cdots, w_n)'$ is an n-dimensional vector of dual variables.

The problem(2. 2) has lower and upper bound constraints. To deal with this bounded variable case, a special linear transformation is employed which is similar to the linear transformation of Vanderbei *et al.*. By defining $D=\text{diag}\,[v_i]$ where $v_i=1-w_i$ if $w_i\geq0$, $v_i=1+w_i$ if $w_i<0$, $i=1, 2, \cdots, n$, and putting $\mathbf{w}=D\mathbf{r}$, we can transform the problem(2. 2) in $\mathbf{w}$-space to the problem(2. 3) in $\mathbf{r}$-space:

$$\text{maximize} \quad (D\mathbf{y})'\mathbf{r}$$

$$\text{subject to } (DX)'\mathbf{r}=\mathbf{0} \qquad (2.3)$$
$$-\mathbf{l}\leq D\mathbf{r}\leq \mathbf{l}.$$

This transformation does not affect the solution of the problem(2.2). The motivation for considering the linear transformation $\mathbf{r}=D^{-1}\mathbf{w}$ is the following: If the current feasible point is located close to the boundary of the polytope, the movement in the gradient direction may not give a substantial reduction in the objective function. Therefore, the current feasible point is linearly transformed to a point which makes the smaller slacks in the inequality constraints so that all of the boundaries are sufficiently distant from that point in the new space. In the transformed space, the current feasible point can be moved sufficiently to a new point along the projected gradient.

At the k-th iteration we compute the projection of $D_k\mathbf{y}$ onto the null space of $(D_kX)'\mathbf{r}_k=\mathbf{0}$ as follows

$$\tilde{e}_k=[I-(D_kX)\{(D_kX)'(D_kX)\}^{-1}(D_kX)'](D_k\mathbf{y}).$$

And then an improved feasible point($\mathbf{r}_{k+1}$) can be obtained by moving along the projected gradient. Denoting λ be a step length we can form the following,

$$\mathbf{r}_{k+1}=\mathbf{r}_k+\lambda\tilde{e}_k.$$

Mapping this new point back to the w-space by the inverse transformation of $\mathbf{w}=D\mathbf{r}$, we get

$$\mathbf{w}_{k+1}=\mathbf{w}_k+\lambda D_k\tilde{e}_k.$$

Let $\mathbf{p}_k=D_k\tilde{e}_k$ and $\lambda=\alpha/\omega_k$ with $0<\alpha<1$, then

$$\mathbf{w}_{k+1}=\mathbf{w}_k+(\alpha/\omega_k)\mathbf{p}_k$$

where $\mathbf{p}_k$ is the projected gradient in the w-space at the k-th iteration.

Since we want to ensure the feasibility of new point($\mathbf{w}_{k+1}$), i.e., $X'\mathbf{w}_{k+1}\mathbf{0}=$ and $-\mathbf{l}\leq\mathbf{w}_{k+1}\leq\mathbf{l}$, ω_k should be chosen as,

$\omega_k=\max_i\{u_{k,i}\}$ where $u_{k,i}=p_{k,i}/(1-w_{k,i})$ if $p_{k,i}\geq 0$, $u_{k,i}=-p_{k,i}/(1+w_{k,i})$ if $p_{k,i}<0$

where $w_{k,i}$ and $p_{k,i}$ are the i-th element of vector $\mathbf{w}_k$ and p_k, respectively. This result is shown in the proof of Theorem 2.1.

This scaling scheme requires fewer iterations and less computation per iteration than transforming the vairables and explicitly including the lower and upper bound constraints within the constraint matrix X in(2.2) by employing slack variables. The proposed algorithm generates a sequence of points, $\hat{\boldsymbol{\beta}}_0, \hat{\boldsymbol{\beta}}_1, \cdots, \hat{\boldsymbol{\beta}}_k$, as follows:

Algorithm BKIML1

(i) Initialization; set $k=0$ and $\mathbf{w}_k=\mathbf{0}$.

(ii) Define $D_k=\text{diag}\,[v_{k,i}]$, where $v_{k,i}=\min\,\{1+w_{k,i},\ 1-w_{k,i}\}$, $i=1,\cdots,n$. Transform X and $\mathbf{y}$ into $\tilde{X}=D_kX$ and $\tilde{y}=D_k\mathbf{y}$, respectively. Compute $\tilde{e}_k=\{I-\tilde{X}(\tilde{X}'\tilde{X})^{-1}\tilde{X}'\}\tilde{y}$.

(iii) Compute $\mathbf{p}_k=D_k\tilde{e}_k$.

(iv) If $\|\mathbf{p}_k\|_\infty<\delta$ for some chosen tolerance $\delta>0$, then compute $\hat{\boldsymbol{\beta}}_k=(\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{\mathbf{y}}$ and stop.

(v) Let $\omega_k=\max_i[\max\{p_{k,i}/(1-w_{k,i}),\ -p_{k,i}/(1+w_{k,i})\}]$. And set the new iterate

$$\mathbf{w}_{k+1}=\mathbf{w}_k+(\alpha/\omega_k)\mathbf{p}_k.$$

Increment k by one and go to step (ii).

We note here that, in our Monte Carlo simulation experiments, $\delta=10^{-6}$ and $\alpha=0.97$ appear to work well in the proposed algorithm.

The computation of $\hat{\boldsymbol{\beta}}_k=(\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{\mathbf{y}}$ in step (iv) can be obtained as follows. From the proof of Proposition 1 in Vandervei *et al.* (1985) and the problem(2.3), it follows that there exists a vector $\boldsymbol{\beta}_k$ such that

$$\boldsymbol{\beta}_k'(D_kX)'=(D_k\mathbf{y})'. \tag{2.4}$$

It can be shown that $\boldsymbol{\beta}_k$ is the vector of dual variables corresponding to the constraint $(D_kX)'\mathbf{r}=\mathbf{0}$ of the problem(2.3). And the scaling leaves the duals with respect to the problem (2.2) unchanged. Since (D_kX) has full rank from the assumption, $(X'D_k^2X)^{-1}$ exists for all k. Hence, (2.4) can be rewritten as

$$\begin{aligned}\boldsymbol{\beta}_k&=(X'D_k^2X)^{-1}X'D_k^2\mathbf{y}\\&=(\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{y}.\end{aligned}$$

It is the vector of current estimates of $\boldsymbol{\beta}$ which is the reweighted least squares solution. This is the main idea of the proposed algorithm which combines the iterative reweighted least squares method and the linear programming approach.

In computing $\hat{\boldsymbol{\beta}}_k=(\tilde{X}'\tilde{X})^{-1}\tilde{X}'\tilde{\mathbf{y}}$ in this algorithm, we notice severe numerical accuracy problems associated with the inversion of $(\tilde{X}'\tilde{X})$, since the matrix $(\tilde{X}'\tilde{X})$ can often be a highly ill-conditioned matrix. Thus we employ an orthogonal decomposition approach, in particular, Householder transformation used in the least squares method. This approach can avoid many numerical problems associated with the computation of $\hat{\boldsymbol{\beta}}_k$ and the projected gradient, and yields accurate and stable solutions even though the matrix

X is highly ill-conditioned. Moreover, at each iteration of the algorithm, the residuals and hence the projected gradient can be computed without explicitly computing $\hat{\boldsymbol{\beta}}_k$. Therefore, this implementation reduces a significant number of computations.

Theorem 2.1. Each iterate $\boldsymbol{w}_k$ of this algorithm is feasible for the problem(2.2).

Proof. In the initial step we set $\boldsymbol{w}_0=\boldsymbol{0}$ as an interior feasible point, therefore $X'\boldsymbol{w}_0=\boldsymbol{0}$ and $-\boldsymbol{l}\leq\boldsymbol{w}_0\leq\boldsymbol{l}$, that is, $\boldsymbol{w}_0$ is feasible. Assume that feasibility holds at the k-th iteration, then

$$\begin{aligned} X'\boldsymbol{w}_{k+1} &= X'\{\boldsymbol{w}_k+(\alpha/\omega_k)\boldsymbol{p}_k\} \\ &= X'\boldsymbol{w}_k+(\alpha/\omega_k)X'[D_k^2\{\boldsymbol{y}-X(X'D_k^2X)^{-1}X'D_k^2\boldsymbol{y}\}] \\ &= X'\boldsymbol{w}_k+(\alpha/\omega_k)\{X'D_k^2\boldsymbol{y}-X'D_k^2X(X'D_k^2X)^{-1}X'D_k^2\boldsymbol{y}\} \\ &= X'\boldsymbol{w}_k \\ &= \boldsymbol{0}. \end{aligned}$$

And ω_k must satisfy the inequality

$$-1\leq w_{k,i}+p_{k,i}/\omega_k\leq 1,\ \text{i.e.,}\ -1<w_{k+1,i}<1.$$

Therefore, it follows that

(i) if $p_{k,i}\geq 0$, then $\omega_k\leq -p_{k,i}/(1+w_{k,i})$ or $\omega_k\geq p_{k,i}/(1-w_{k,i})$

(ii) if $p_{k,i}<0$, then $\omega_k\leq p_{k,i}/(1-w_{k,i})$ or $\omega_k\geq -p_{k,i}/(1+w_{k,i})$.

Consequently, we can choose

$$\omega_k=\max_i[\max\{p_{k,i}/(1-w_{k,i}),\ -p_{k,i}/(1+w_{k,i})\}].$$

Thus, the feasibility holds at each iteration. ▨

Corollary 2.1. Weak duality holds between the problem(2.1) and the problem(2.2).

Proof. From the feasibility of $\boldsymbol{w}_k$ in Theorem 2.1, it follows that

$$\begin{aligned} \boldsymbol{y}'\boldsymbol{w}_k &= \boldsymbol{e}_k'\boldsymbol{w}_k \\ &\leq \|\boldsymbol{e}_k\|_1\cdot\|\boldsymbol{w}_k\|_\infty \\ &\leq \|\boldsymbol{e}_k\|_1 \text{ since } \|\boldsymbol{w}_k\|_\infty\leq 1 \\ &= \boldsymbol{l}'\boldsymbol{e}^+ + \boldsymbol{l}'\boldsymbol{e}^-. \end{aligned}$$ ▨

Lemma 2.1. If $\boldsymbol{p}_k=\boldsymbol{0}$, then $\boldsymbol{w}_k$ is optimal. Otherwise assume that $\boldsymbol{p}_k\neq\boldsymbol{0}$ for all k. Then $\{\boldsymbol{y}'\boldsymbol{w}_k\}$ converges in problem(2.2).

Proof. If $\boldsymbol{p}_k=\boldsymbol{0}$, then $\tilde{e}_k=\boldsymbol{0}$ which means that the objective function is a constant for all feasible solution. Hence, in particular, $\boldsymbol{w}_k$ is optimal. Now suppose $\boldsymbol{p}_k\neq\boldsymbol{0}$ for all k, and first show that $\{\boldsymbol{y}'\boldsymbol{w}_k\}$ is a strictly increasing sequence. The projected gradient in the transformed space is obtained as

$$\tilde{e}_k=D_k\boldsymbol{y}-D_kX\{(D_kX)'(D_kX)\}^{-1}(D_kX)'(D_k\boldsymbol{y})$$

and it is transformed to the original space by,

$$p_k = D_k^2 y - D_k^2 X\{(D_k X)'(D_k X)\}^{-1}(D_k X)'(D_k y). \tag{2.5}$$

It follows from equation(2.5) that

$$[y - (D_k^2)^{-1} p_k]' = y' D_k^2 X (X' D_k^2 X)^{-1} X'. \tag{2.6}$$

Now the follwing relationship can be obtained from Theorem 2.1 and (2.6),

$$X'(w_{k+1} - w_k) = \mathbf{0}$$

$$y' D_k^2 X (X' D_k^2 X)^{-1} X' (w_{k+1} - w_k) = \mathbf{0}$$

$$y'(w_{k+1} - w_k) = p_k'(D_k^2)^{-1}(w_{k+1} - w_k).$$

Furthermore, since $(D_k^2)^{-1}$ is positive definite and $p_k \neq \mathbf{0}$ it follows that

$$\begin{aligned} p_k'(D_k^2)^{-1}\ (w_{k+1} - w_k) &= p_k'(D_k^2)^{-1}\{w_k + (\alpha/\omega_k) p_k - w_k\} \\ &= (\alpha/\omega_k) p_k'(D_k^2)^{-1} p_k \\ &> 0. \end{aligned}$$

Therefore, $y'w_{k+1} > y'w_k$. Since $\{y'w_k\}$ is bounded above by the weak duality in Corollary 2.1, $\{y'w_k\}$ converges. ▨

Theorem 2.2. The proposed algorithm terminates in a finite number of iterations for some chosen tolerance $\delta > 0$.

Proof. From Lemma 2.1, we know that

$$\begin{aligned} y'w_{k+1} - y'w_k &= (\alpha/\omega_k) p_k'(D_k^2)^{-1} p_k \\ &= (\alpha/\omega_k) \|D_k^{-1} p_k\|_2^2. \end{aligned}$$

The convergence of $\{y'w_k\}$ implies that its difference sequence tends to zero,

$$(\alpha/\omega_k) \|D_k^{-1} p_k\|_2^2 \to 0. \tag{2.7}$$

From step(v) of the proposed algorithm it follows that

$$\begin{aligned} 0 \leq \omega_k &= \max_i [\max\{p_{k,i}/(1 - w_{k,i}),\ -p_{k,i}/(1 + w_{k,i})\}] \\ &\leq \max_i \{|p_{k,i}|/v_{k,i}\} \\ &= \|D_k^{-1} p_k\|_\infty \end{aligned}$$

and hence we know that

$$\begin{aligned} 0 \leq \alpha \|D_k^{-1} p_k\|_2 &= \alpha \|D_k^{-1} p_k\|_2^2 / \|D_k^{-1} p_k\|_2 \\ &\leq \alpha \|D_k^{-1} p_k\|_2^2 / \|D_k^{-1} p_k\|_\infty \\ &\leq (\alpha/\omega_k) \|D_k^{-1} p_k\|_2^2. \end{aligned}$$

Therefore, (2.7) implies that $\|D_k^{-1} p_k\|_2 \to 0$. Since

$$\begin{aligned} \max_i |p_{k,i}| &= \|p_k\|_\infty \\ &\leq \|D_k^{-1} p_k\|_\infty \\ &\leq \|D_k^{-1} p_k\|_2 \end{aligned}$$

it follows that $p_{k,i} \to 0$ for all $i=1, \cdots, n$. Hence, the proof is complete. ▨

Example 2.1. We now illustrate the steps of the proposed algorithm on the computer-generated data. This set of data is generated by Hoffman and Shier(1980) test problem generator. There are 7 observations and 3 independent variables including a intercept term. The 2nd and 3rd column of X are generated from a normal distribution with mean (5, 10) and variance(1.3, 1.9), respectively. The optimal L_1 norm estimates of the regression coefficients are specified as(2, −2, 5). The errors were generated from a normal distribution with mean 0 and variance 2.3.

$$X=\begin{pmatrix} 1.000000 & 5.766515 & 9.235767 \\ 1.000000 & 4.661123 & 11.439430 \\ 1.000000 & 2.970308 & 9.238118 \\ 1.000000 & 2.740973 & 11.706110 \\ 1.000000 & 6.769230 & 9.862975 \\ 1.000000 & 4.075700 & 7.034439 \\ 1.000000 & 4.157894 & 12.830360 \end{pmatrix} \qquad y=\begin{pmatrix} 38.55223 \\ 49.57025 \\ 45.27223 \\ 55.04866 \\ 37.77638 \\ 25.13447 \\ 57.83601 \end{pmatrix}$$

The L_1 norm estimates of the parameters in the linear regression model(1.1) are computed in 6 iterations by the proposed algorithm BKIML1. Table 2.1 represents the iterative sequences of the values computed at each iteration.

Table 2.1. A Summary of Each Iteration of BKIML1

itera.	initial	1st	2nd	3rd	4th	5th	6th
	−1.222812	3.415632	1.899648	2.005962	1.999910	1.999998	2.000001
$\hat{\beta}_k$	−1.981091	−2.022338	−1.999285	−2.000271	−2.000001	−2.000009	−2.000009
	5.318323	4.883875	5.005493	4.999597	5.000007	5.000003	5.000003
	2.083245	0.640561	0.022162	0.018250	0.008792	0.000264	0.000234
	−0.811465	−0.218066	−0.137289	−0.004640	−0.004027	−0.002238	−0.000067
	3.248198	0.082373	0.080605	0.034147	0.001025	0.000909	0.000551
$\tilde{e}_k$	−0.555336	0.004152	0.028612	−0.000491	0.000023	0.000005	−0.000000
	−0.044833	−0.117521	0.033490	−0.000187	−0.000002	0.000006	0.000000
	−2.979799	−0.484008	−0.112598	−0.023387	−0.007846	−0.001054	−0.000573
	−0.940010	0.120244	0.022861	0.000328	0.000007	0.000005	−0.000000
	2.083245	0.242059	0.000251	0.000175	0.000041	0.000000	0.000000
	−0.811465	−0.165223	−0.069671	−0.000071	−0.000053	−0.000016	−0.000000
	3.248198	0.002471	0.002117	0.000386	0.000000	0.000000	0.000000
p_k	−0.555336	0.003463	0.024017	−0.000486	0.000023	0.000005	−0.000000
	−0.044833	−0.115947	0.027161	−0.000187	−0.000002	0.000006	0.000000
	−2.979799	−0.053314	−0.003312	−0.000141	−0.000016	−0.000000	−0.000000
	−0.940010	0.086490	0.019438	0.000324	0.000007	0.000005	0.000000

w_k	0.000000	0.622114	0.988663	0.990439	0.995396	0.999862	0.999878
	0.000000	−0.242324	−0.492522	−0.984776	−0.986782	−0.992655	−0.999780
	0.000000	0.970000	0.973742	0.988696	0.999661	0.999699	0.999818
	0.000000	−0.165838	−0.160594	0.009095	−0.004717	−0.002149	−0.000028
	0.000000	−0.013388	−0.188967	0.002939	−0.002361	−0.002625	−0.000030
	0.000000	−0.889849	−0.970582	−0.993985	−0.997981	−0.999729	−0.999853
	0.000000	−0.280713	−0.149741	−0.012408	−0.003214	−0.002403	−0.000060
$\|e_k\|_1$	10.662885	9.413973	9.224974	9.123723	9.122742	9.122725	9.122708
$y'w_k$	0.000000	7.651545	8.752752	9.042275	9.101018	9.118243	9.121284

This example not only illustrates the performance of the proposed algorithm but shows numerical aspects of the theorems, corollary and lemma proven in this paper.

3. Computational Comparison

Some computational experiments were conducted to get an idea of how well the proposed algorithm performs on large data sets. Five algorithms were compared on the basis of computational efficiency, and numerical accuracy and stability of the estimates computed.

The test problem generator, L1GNR, of Hoffman and Shier(1980) was used to generate 25 replicated data sets for each possible combination of number of parameters and number of observations(m=2, 5, 10, 15, 20, 50, 100, 200; n=30, 50, 100, 200, 400). Each data set was generated randomly with specified optimal solutions. It was found that a large number of design matrices had large condition numbers. All computations were done on an IBM 370/158 computer using the FORTVCG compiler. The performance indicators considered were CPU time and number of iterations to compare computational efficiency, and mean absolute deviations and standard errors of the estimates to compare numerical accuracy and stability.

We collected the CPU time(in milliseconds) and the number of iterations for each combination and for each algorithm. The relationships between the CPU time(and the number of iterations) and problem sizes were analyzed using a log-linear regression model to determine the combined effects of variations in m and n on the CPU time(and the number of iterations). Table 3.1 gives the results obtained.

These results show that CPU times of all algorithms are more sensitive to increases in m than to increases in n, and that numbers of iterations, except for the proposed

Table 3.1. Performance Comparison of Algorithms

Algorithm	CPU time	Iteration
Abdelmalek	$0.000056\ m^{2.1493}\ n^{1.7852}$	$0.1161\ m^{1.2816}\ n^{0.6082}$
Armstrong *et al.*	$0.000053\ m^{2.5174}\ n^{1.4953}$	$0.1916\ m^{0.9990}\ n^{0.5369}$
Barrodale and Roberts	$0.000540\ m^{1.6580}\ n^{1.4370}$	$0.8259\ m^{1.0968}\ n^{0.1952}$
Bartels *et al.*	$0.002596\ m^{1.7416}\ n^{1.2183}$	$0.3623\ m^{1.2741}\ n^{0.2995}$
Kim	$0.004129\ m^{1.5118}\ n^{1.2210}$	$5.2571\ m^{0.0314}\ n^{0.1090}$

algorithm, are also more sensitive to increases in m than to increases in n. We note that the CPU time and the number of iterations of the proposed algorithm are the least sensitive to increases in problem size.

Along with comparing the computational efficiency of algorithms, we checked numerical accuracy and stability through the mean absolute deviations and the standard errors of estimates computed by five algorithms. There was no significant difference in numerical accuracy among the algorithms. But the proposed algorithm computed slightly more accurate estimates than the others, and yielded the most stable ones.

In the light of these experiments, the proposed algorithm appears to be more computationally efficient than the others, except for Barrodale and Roberts', when problem size is considerably large. In addition, it yields numerically accurate and stable estimates irrespective of problem size.

4. Concluding Remarks

A computational procedure based on a modification of Karmarkar algorithm for linear programming problem is proposed for minimum L_1 norm estimation. The proposed algorithm yields numerically accurate and stable estimates even though design matrix X is highly ill-conditioned. It is achieved by employing the orthogonal decomposition approach.

It can be shown that the proposed algorithm requires fewer iterations to reach the optimal solution than are required by the simplex-type algorithms when problem size is large. The reason is that our algorithm cuts across the interior of a convex region while the simplex-type algorithms move from vertex to vertex of a convex polyhedral region. (But some algorithms use the bypassing technique).

An additional advantage of this algorithm is that it is computationally simple and easy to implement because any basic least squares algorithm, an accessible part of any subroutines or statistical packages, can be used.

On the other hand, since the amount of computation required per iteration is usually greater for the proposed algorithm, it is not the most computationally efficient algorithm. One important point to be considered is the updating of the orthogonal decomposition in the implementation of this algorithm. If we can update the previous orthogonal decomposition of $\tilde{X}=D_kX$ at the next iteration, this would significantly reduce the number of computations required by the proposed algorithm.

References

(1) Abdelmalek, N.N. (1980). L_1 Solution of Overdetermined Systems of Linear Equations, *ACM Transactions on Mathematical Software*, Vol. 6, 220~227.

(2) Armstrong, R.D., Frome, E.L. and Kung, D.S. (1979). A Revised Simplex Algorithm for the Absolute Deviation Curve Fitting Problem, *Communications in Statistics-Simulation and Computation*, B8, 175~190.

(3) Barrodale, I. and Roberts, F.D.K. (1973). An Improved Algorithm for Discrete L_1 Linear Approximation, *SIAM Journal of Numerical Analysis*, Vol. 10, 839~848.

(4) Bartels, R.H., Conn, A.R. and Sinclair, J.W. (1978). Minimization Techniques for Piecewise Differentiable Functions; The L_1 Solution to an Overdetermined Linear System, *SIAM Journal of Numerical Analysis*, Vol. 15, 224~241.

(5) Bassett, Jr. G. and Koenker, R. (1978). Asymptotic Theory of Least Absolute Error Regression, *Journal of the American Statistical Association*, Vol. 73, 618~621.

(6) Blattberg, R. and Sargent, T. (1971). Regression with Non-Gaussian Stable Disturbances; *Some Sampling Results*, *Econometrica*, Vol. 39, 501~510.

(7) Dielman, T. and Pfaffenberger, R. (1982). LAV(Least Absolute Value) Estimation in Linear Regression; a Review, *TIMS/Studies in the Management Sciences*, Vol. 19, 31~52.

(8) Fisher, W.D. (1961). A Note on Curve Fitting with Minimum Deviations by Linear Programming, *Journal of the American Statistical Association*, Vol. 56, 359~362.

(9) Hoffman, K.L. and Shier, D.R. (1980). A Test Problem Generator for Discrete Linear L_1 Approximation Problems, *ACM Transactions on Mathematical Software*, Vol. 6, 587~593.

(10) Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming, *Combinatorica*, Vol. 4, 373~395.

(11) Kiountouzis, E.A. (1973). Linear Programming Techniques in Regression Analysis, *Applied Statistics*, Vol. 22, 69~73.

(12) Pfaffenberger, R.C. and Dinkel, J.J. (1978). Absolute Deviations Curve Fitting; An Alternative to Least Squares, Contributions to *Survey Sampling and Applied Statistics*, edited by H.A. David, Academic Press, NY, 279~294.

(13) Rice, J.R. and White, J.S. (1964). Norms for Smoothing and Estimation, *SIAM Review*, Vol. 6, 243~256.

(14) Rosenberg, B. and Carlson, D. (1977). A Simple Approximation of the Sampling Distribution of Least Absolute Residuals Regression Estimates, *Communications in Statistics-Simulation and Computation*, B6, 421~437.

(15) Schlossmacher, E. J. (1973). An Iterative Technique for Absolute Deviations Curve Fitting, *Journal of the American Statistical Association*, Vol. 68, 857~859.

(16) Vanderbei, R. J., Meketon, M. S. and Freedman, B. A. (1985). A Modification of Karmarkar's Linear Programming Algorithm, Working Paper, AT & T Bell Laboratories, Holmdel, NJ.

(17) Wagner, H. M. (1959). Linear Programming Techniques for Regression Analysis, *Journal of the American Statistical Association*, Vol. 54, 206~212.

(18) Wesolowsky, G. O. (1981). A New Descent Algorithm for the Least Absolute Value Regression Problem, *Communications in Statistics Simulation and Computation*, B10, 479~491.